

REVIEWS: LIGHTWAVE 7B, SPECTRA FOO, ACCUREV & MORE

eDevelor

LINE OF GAME INNOVATION

Confront Your Memory Demons

Tips For Effective Memory Management

How To Use Symbols To Evoke **Emotion In Games**

Secrets Of A Successful Launch

Mythic's DARK AGE OF CAMELOT

հIII ու հահահահեն ապահականի հայտես հանձեն անձե

0247

Display Until February 22, 2002

2002.20



Built for production by House of Moves.

www.moves.com

Game Developers Conference

Conference: March 19-23, 2002 Expo: March 21-23, 2002

San Jose, CA

Conference Brochure

Make better games

Learn directly applicable techniques in the core disciplines of game development.

Tracks:

Visual Arts

Game Design

Programming

Audio

Business & Legal

Level Design

Production

www.gdconf.com



Dear game developer,

Join your colleagues at the Game Developers Conference and you'll go home with new ideas, new skills, and the ability to implement them. At GDC 2002, you'll meet delegations from Europe, Asia, the United States and the rest of the world on the same quest: to make better games. See you in San Jose.

Best Regards,

Alan Yu Director

Game Developers Conference

Table of Contents	
Keynotes	3
Expo & Exhibitors	4
Featured Events	5
Wireless Game Summit	5
Tutorials	6
Sessions	12
Events	18
Schedule-at-a-Glance	19
Advisory Board	20
Travel	21
Pass Options	21
Registration	22

Cinemati	ic Lighting	for	3D	Graphics
Jaramy F	Rirn rando	r34	con	_

- **DEPISORIE ENTERTAINMENT: Viva La Evolution!**Neil Young, Electronic Arts
- Fur Using Shells and Fins
 Jerome E. Lengyel, Microsoft Research
- The Game Design of VIB RIBBON 2

 Masaya Matsuura, creator, PaRappa THE Rapper
- Games with Collectible Components
 Richard Garfield, Wizards of the Coast
- Global IP Protection and Anti-Piracy Techniques
 Chris Chapin, Electronic Arts
- High Dynamic Range Imagery and Image-Based Lighting
 Paul Debevec, USC Institute for Creative Technologies
- ▶ The Making of SHREK Nick Walker, PDI/Dreamworks
- Manhattan as Muse: New York City as a Conceptual Tool Duncan Brown, ZenLux
- Maximizing DMA Bandwidth on the PlayStation2
 Greg Omi, Naughty Dog

conference highlights

- Mobile Phone Games: Where Are They Now and Where Are They Going? Ian Baverstock, Kuju Entertainment
- Patents: Are Things Getting Better? Tim Wu, Riverstone Networks
- Quality or Death: How the Business Model and Player Psychology Collide in For-Pay Online Games Eric Todd. Maxis
- Rez: The Synesthesia that Games Invite
 Tetsuya Mizuguchi, creator, SPACE CHANNEL 5
- SEAMAN: The Case for Eccentric Title Making Yoot Saito, Vivarium
- This Is Not a Game: A Discussion of the Creation of the Al Web Experience

 Elan Lee & Jordan Weisman, Microsoft
- Toss it in the Blender:
 Nonlinear Animation for Games
 Steve Theodore, Valve Software
- Vector Units and Quaternions
 Jim Van Verth, Red Storm Entertainment
- What Worked: 10 Detailed Examples of Successful Character Action Design Mark Cerny, Cerny Games
- When Maps Collide: A Conversation with Will Wright and Scott McCloud
- Wireless Game Design Review David Collier, PacketVideo

make better games

Learn about successful game creation from the world's leading developers. Keynotes, in-depth tutorials, and more than 300 lectures, panels and roundtables cover every aspect of game development for all platforms and all genres.

keynotes



Visual Arts Peter Chan Conceptual Illustrator HARRY POTTER AND THE SORCERER'S STONE

Chan has worked on games (DAY OF THE TENTACLE, FULL THROTTLE, GRIM FANDANGO, PUTT PUTT, PAJAMA SAM, VAMPIRE: THE MASQUERADE, to name a few), films, and advertising since 1989. After 4 years at LucasArts, he moved to a remote island in Washington State to launch his freelance career (a giant and illogical leap of faith initiated by quality of life issues). Film credits include ANTZ, STAR WARS: EPISODE 1 THE PHANTOM MENACE, MONKEYBONE, and HARRY POTTER AND THE SORCERER'S STONE. He recently completed two products for the Xbox (New Legends for Infinite Machine and an unannounced Tim Schafer game).



Business & Legal Gabe Newell Co-founder, Valve Software

Gabe Newell is the founder and managing director of Valve, an entertainment software company based

in Kirkland, Washington. Valve's first game, HALF-LIFE, has won more than 50 Game of the Year honors, and was called "Best Game of All Time" by PC Gamer. Before starting Valve, Gabe held a number of positions at Microsoft, where his responsibilities included running program management for the first two releases of Windows, and starting the company's multimedia division. Valve is currently working on TEAM FORTRESS 2: BROTHERHOOD OF ARMS.



Programming John H. Conway Professor of Mathematics, Princeton University John Horton Conway is a prolific mathematician active in the theory of finite groups, knot theory,

number theory, game theory and coding theory. Among amateur mathematicians, he is perhaps most widely known for his mathematical games, in particular the creation of the $\ensuremath{\mathsf{GAME}}$ of Life. He invented a new number system, the surreal numbers, which are closely related to certain games and have been the subject of a mathematical novel by Donald Knuth.

Audio

Ray Colcord

President, The Society of Composers and Lyricists

Ray Colcord is a former A & R executive and producer with such companies as Columbia Records and United Artists records. Ray produced Aerosmith's second album and toured as a keyboard player with Lou Reed and Don Maclean. He is the president of The Society of Composers & Lyricists and the founder and President of the Composers and Lyricists Legal Aid Society. Ray's television and film scores credit include Boy MEETS WORLD, BIG BROTHER 2, DINOSAUR, THE SIMPSONS, DUMB & DUMBER, DOUBLE IMPACT, and DEEP IMPACT. He has also taught music production for Film and Television for the UCLA extension school.



Production Shuhei Yoshida VP, Product Development, Sony Computer Entertainment America

Shuhei Yoshida, a 15-year Sony veteran and renowned software developer, has been with the PlayStation group since the project's inception more than eight years ago. Through his tenure with Sony Computer Entertainment Inc. in Japan, he held various high-profile positions in both the third party relations and product development arenas. In April 2000, Shuhei was appointed vice president, product development, Sony Computer Entertainment America Inc. and is responsible for all product development activities for Sony Computer Entertainment America Inc. and the 989 Sports brand.



General Interest Shin'ichi Okamoto

CTO, Vice President, Sony Computer Entertainment Inc. Shin'ichi Okamoto joined Sony Music Entertainment Inc., where he began his career with games, launch-

ing and organizing a game development team in 1983. Shin'ichi was enlisted in Sony's computer entertainment project in 1992, which soon took the form of Sony Computer Entertainment Inc., where he managed three engineering units as the director of development, one to provide an OS, another a development environment and lastly, a technical support group for the PlayStation. In April 1999, Shin'ichi became the senior vice president of R&D, playing a key role in the development of PlayStation2. In April 2001, Shin'ichi was assigned to the position of chief technology officer supervising R&D and the Broadband and Business Division.



Level Design Toshihiro Nagoshi President, Amusement Vision

Toshihiro Nagoshi graduated from the Tokyo University of Art and Design in 1989 and joined

Sega shortly thereafter, working for AMII under Yu Suzuki as a CG designer. He quickly ascended the ranks, and in 1999 became the manager of AMII. When AMII became Amusement Vision, Toshihiro took the helm as president. Recently, Toshihiro has worked on the game designs for Super Monkey Ball, Viruta Striker 3 and DAYTONA USA 2001.

Game Design

Kazunori Yamauchi President, Polyphony Digital

Kazunori Yamauchi is the producer and director of GRAN TURISMO, the hit car racing simulation game series that has shipped over 23 million units worldwide. He is also president of Polyphony Digital Inc. Currently Kazunori is committed to the production of GRAN TURISMO CONCEPT 2001 TOKYO for PlayStation2, which is expected to be released on January 1, 2002.

evaluate new products & technologies

Interactive entertainment is driving technology and creating the future. The GDC is where the game industry meets to evaluate new products, learn about technological breakthroughs, and define the next generation. Over 200 exhibitors demonstrate the technology enabling developers to make better games.

Expo Hours

Thursday, March 21 11:30am-7:30pm Friday, March 22 11:30am-7:30pm Saturday, March 23 11:30am-4pm

3D Labs
3D Pipeline
Absolute Quality
Advanced Micro Dev

Advanced Micro Devices (AMD)

AK Peters
Alias|Wavefront
Analog Devices
Angel Studios
Animation magazine
ATI Technologies Inc

Biovirtual
Blizzard Entertainment

CG² Inc

Charles River Media
Computer Graphics World

Criterion Software

Develop

Digital Artist Management Inc Digital Eclipse Software Inc

Discreet Dolby Labs DTS

Electronic Arts

Elsa Ensemble Studios

Ensemble Studios
Epic Games
Essential Reality
Eyetronics
Gamasutra.com

Game Developer magazine

Gamespy Industries

Gigex Gignews.com Guillemot Inc Havok

House of Moves ID8 Media

IGDA

Immersion Corporation

Infogrames Inc InSpeck IntegrityWare Intel

Interact
Intrinsic Graphics

Konami of America Inc

Left Field

Lifemode Interactive

Lithtech Logitech

LucasArts Entertainment

Macrovision State of Maryland

MCV Mathengine Metrowerks

Microboards Technology

Microsoft
Microsoft Xbox
Midway Games West Inc
Monolith Productions
Morgan Kaufmann
Namco Hometek Inc

NewTek

Nintendo of America Numerical Design Ltd

Nvidia
NxN Software
Oddworld Inhabitants
Packet Video Corporation
Perforce Software
Premier Press
Premier Search Inc
Prime Candidate

Qualcomm Quazal

RAD Game Tools Rainbow Studios Real Networks Rebel Arts

Red Storm Entertainment Right Hemisphere Saffire Corporation Scientific Placement

Scientific Placement Sega of America Singular Inversions

SiS

SN Systems Softimage Sony Computer

Entertainment America
Sony Online Entertainment Inc

Sound Ideas Soundball

ST Microelectronics Stormfront Studios Studio Search Inc Sun Microsystems Inc.

Synovial Inc Technicolor Testing Testing 123

IIIQ

Turbine Entertainment Software Turbo Squid Inc Vicarious Visions Vicon Motion Systems Viewpoint Corporation Virtual Search Inc Vivendi Universal

Wordware Publishing

Zona

Exhibitors as of December 7, 2001. Go to **www.gdconf.com** for updates.

wireless game summit

SATURDAY MARCH 23

Wireless Games: The Current and Future Landscape of Wireless

Andrew Seybold & Barney Dewey

Wireless operators in the United States tried and failed to interest consumers in wireless data. Now, they have regrouped and are moving forward on a two-pronged approach. One area of renewed focus is providing wireless data access to the business world; the second is providing content such as games that appeal more directly to consumers. Meanwhile, games are alive and well in Asia. Korean network operators are now offering wireless network speeds of 100 Kbps and in Japan, NTT DoCoMo has launched their own 3G wireless network. This seminar focuses on the current state of the industry for wireless games and looks into the not-too-distant future.

The session starts with the bottom line:

How do you make money writing games for wireless devices today and in the future; who will control content distribution; what capabilities do the networks have today and what will they have tomorrow?

Issues covered include:

What networks are capable of packet data services today? What networks will be on line within the next year? Data speeds are different in Europe, Asia, and in the United States. It is important to understand just how different. Which of the wireless operators have the most users that fit into the gamer profile? In Japan, where more than 30 million wireless consumers have data access, games have become a mainstay of the wireless operators' offerings. The business models in the United States are evolving; the wireless operators are set in fierce competition for customers, and are trying to differentiate themselves from the pack. How do games play in the US model today and how will their evolution over time impact content providers? Will a single type of device, a single programming language, and a single network technology rule the future? Today, there are multiple digital wireless standards, scores of different wireless devices, and at least a half-dozen different operating systems that must be considered by the developer community. How do you sort this all out, or do you just wait for a "winner" to be crowned?

This session discusses the different wireless platforms: phones, two-way pagers, PDAs, smartphones and more. It explores the different languages developed specifically for wireless devices: J2ME, BREW, WAP, and other standards such as the ones being pushed by Texas Instruments, Intel, and ARM itself. Other questions addressed include: How can I make money with wireless games? How do I get my products into the market? Do I have to work with all of the wireless operators both here and abroad or can I leverage others in sales and marketing?

Intended audience:

Designed for producers, publishers and developers responsible for business strategy, the Wireless Game Summit assesses the state of the industry today, looks ahead to the future and tackles the questions the leaders in mobile game content will need to answer to succeed in this emerging segment.

Andrew Seybold is considered to be one of the premier industry analysts within the mobility and wireless communications industry. Andrew currently heads the Andrew Seybold Group, LLC and is editor-in-chief of Forbes/Andrew Seybold's Wireless Outlook. He served as vice president and chief analyst for Dataquest's mobility practice and in a number of technical and management capacities with Motorola, General Electric, and RCA. Andrew is a co-founder and permanent board member of the Portable Computer and Communications Association and he is a member of the IEEE, APCO, ARRL, and a Fellow in the Radio Club of America.

Barney Dewey has been involved on both the communications and computing industries for thirty years. In 1997, he joined the Andrew Seybold Group, LLC, a highly regarded consulting firm specializing in the connected mobility space where mobile computing and advanced communications technology meet. He also joined Andrew Seybold's Outlook, Inc., a newsletter and conference company. In July 2000, Barney and Seybold launched Forbes/Andrew Seybold's Wireless Outlook, a monthly newsletter published jointly with Forbes. Prior to joining the Andrew Seybold Group, Barney was responsible for the communications and connectivity strategy and products of Apple's Newton handheld computer. Barney spent ten years with Motorola designing and implementing wireless communications systems

featured events



Celebrate the developers of the best games of 2001.

Presented by the IGDA, the Game Developers Choice Awards are where the game industry pays tribute to the art, effort

and sacrifices that go into making truly innovative interactive entertainment. www.igda.org/awards Presented by:

Sponsored by:





INDEPENDENT GAMES FESTIVAL

Experience unsigned games that break genres, push the boundaries of gameplay and redefine the interactive experience. Play the finalist games at the IGF Pavilion, and see this year's winners at the Game Developers Choice Awards ceremony.

The Independent Games Festival inspires and empowers the game development community by recognizing innovative games from around the world and connecting independent talent with publishing resources.

www.igf.com

Principal Sponsor:

Craft Sponsor:





GameDevelopers See the year's best Conference real-time reel

real-time game art clips. The GDC Real-

Time Reel celebrates the beauty of real-time visuals.

Entries come from all areas of computer graphics and interactive techniques as they relate to game development and are considered by a selected jury representing a broad range of expertise in game development and interactive techniques. All submissions are juried together based on content, technical implementation, innovation, artistic achievement, production values, creativity, originality, narrative, design, entertainment value, and advancement of visualization in the artistic field of game development.

Enter your clip by February 1, 2002. www.gdconf.com/reel

focus on a critical topic

Intensive one and two day sessions go in-depth on the essentials of successful game development. GDC tutorials are solutions-oriented and provide a comprehensive understanding of relevant issues and techniques.

TWO DAY TUTORIALS TUESDAY, MARCH 19 & WEDNESDAY, MARCH 20

150

GAME TUNING WORKSHOP

Marc LeBlanc

Game design is an iterative process. In any medium, the final quality of a game depends on many cycles of experimentation, analysis and revision, also known as tuning. This workshop focuses on the tuning process, presenting a formal approach to game design in which games are viewed as systems and analyzed using formal models.

Before we can even begin to design a game, we need to understand our aesthetic goals. In other words, we need to enumerate all the kinds of "fun" that we hope the game will provide its users. We can formalize our understanding of our game's aesthetic goals by formulating an aesthetic model for each goal, a formal description of the goal that identifies its criteria for success and possible modes of failure.

This workshop presents a handful of aesthetic models as examples and encourages attendees to formulate their own. Attendees use game design exercises to practice using aesthetic models as a yardstick to measure their progress through the tuning process. The workshop begins by challenging attendees to work in small groups to analyze a popular digital game in terms of its aesthetic goals and models. Having gained some practice with this kind of analysis, attendees are asked to analyze a handful of games drawn from various media. Several different games are explored, and common game design themes are identified as different groups share their results.

For each game they analyze, attendees are presented with a concrete design exercise to undertake. An exercise might involve adding a new feature, accommodating a new goal or requirement, or fixing a design flaw. These exercises challenge attendees to analyze and identify the design principles at work in a game, and to think flexibly and creatively while working within design constraints. In addition, these exercises serve as a starting point for discussing how the tuning process applies to games in digital and non-digital media. In addition to these analysis-andrevision exercises, attendees gain further practical experience working with these models through brief collaborative design projects, brainstorming sessions, critical analysis, and discussion.

Marc LeBlanc worked for eight years at Looking Glass Studios, and was a core contributor to several award-winning PC game titles, including the THIEF and SYSTEM SHOCK series. His roles have included project leader, programmer, and game designer. Caught in the blast radius of Looking Glass' explosive demise last May, Marc finally touched down at Visual Concepts Entertainment, where he is currently hard at work developing technology for their next round of titles.

151 IGDA ACADEMIC SUMMIT

presented by the IGDA

The last five years have seen a steep increase in the complexity of the game production process. Larger teams, budgets, technical issues, and project scope have generated unprecedented interest in more effective analytical tools and management techniques. At the same time, universities have begun to take note of games as a cultural force and as a potential career path for students interested in the burgeoning game field. While industry and academia are more involved with each other than in the past, there is room to improve communication, actively work on each other's problems, and push forward both the game industry and university curriculum. Ties with academia can help foster the growth and development of the industry by ensuring graduates come to the job market with a solid knowledge base. Conversely, the academic environment supports analysis and examination of the medium in a way the purely commercial industry has trouble with, providing space for formal and conceptual experimentation that is hard to justify in the market-driven world of shipping titles.

The Education Committee of the International Game Developers Association hosts this two-day summit in order to bring together pioneering members of the academic community with developers interested in education. Together, the groups tackle issues of research initiatives and relations, game design and development curriculum, and other topics relevant to the common agenda.

Intended audience: All academics and developers interested in building stronger relations with each other are encouraged to attend. From the academic world, professors, department heads, faculty directors and other academic decision-makers should attend this session. From industry, developers and managers who want to collaborate with academia on research projects, build stronger relationships, and help guide curriculum are encouraged to attend.

ONE DAY TUTORIALS DAY ONE - TUESDAY, MARCH 19

101 CHARACTER DESIGN AND DIGITAL PAINTING

Don Seegmiller

This tutorial discusses and demonstrates techniques that can be used when designing and digitally painting a character. This information is useful to anyone who develops character designs and needs to present ideas in something more than a pencil sketch, as well as to those who just want to be able to create digital works of art.

Takeaway: Attendees leave with new ideas on how to increase their creativity and with new techniques to express those ideas. They also gain a greater appreciation of the potential of current digital painting programs.

Don Seegmiller has been art director of Saffire for the last five years. Don is directly responsible for managing 40 artists at both Saffire locations. Occasionally, if he is lucky, he actually gets to do some art. His art has been featured in Spectrum 7, Design Graphics Magazine, and Step by Step Electronic Design and the Painter 6 Wow Book

102 3D Studio Max

Pia Maffei

This tutorial introduces users to animation and modeling techniques for the game industry using 3D Studio Max and Character Studio. Topics discussed for 3D Studio Max include interface customization, modeling options for both high and low polygon counts, and IK rigging and animation. Advanced animation techniques including animation layering and crowd systems with Character Studio are covered.

Pia Maffei has been using 3D Studio Max since its inception. Today, she trains new and experienced Max users at California State University at Long Beach, other Discreet-certified training facilities, and game companies throughout the US and Canada.

103 Advanced Mapping with Precision

Tito Pagan

This tutorial covers the custom texture creation and mapping process for real-time game art, addressing practical and theoretical issues, tricks and techniques and production art paths as they apply to both 3D Studio Max and Maya.

Takeaway: Attendees receive training in the precise application of custom textures for both organic and non-organic models.

Tito Pagan is a veteran 3D artist for real-time games with 10 years of industry experience. He has created art for 13 PC game titles and two Internet games, including ENCARTA 2002 by Microsoft, DUNGEON SIEGE by Gas Powered Games, and BEITY BAD, a browser-based Internet game by WildTangent.

104 ADVANCED DIRECT3D

Sim Dietrich & Richard Huddy

This one-day course is aimed at developers working with Direct3D using the DirectX 8 or 9 interfaces. The course teaches the handling of modern high-performance accelerators to achieve both high-performance and high-quality results. There are six principal sections to the course:

- Pure performance: The measurement and pursuit of software efficiency, what tools can help and when; the effects of use and abuse of the Direct3D API; and why most games are still CPU-bound and what you can do to change that.
- **Scalability:** As the breadth of features available in consumers' machines increases, strategies for exploiting the diverse resources continue to increase in importance.
- Tricks and traps: An assortment of ideas collected from broad experience, including those that work and how to build on them, and those that don't and how to avoid the same mistakes.
- Shadows: A look at techniques for making high-quality shadows with high-performance, from those that scale well from low-end hardware to those that expose the newest DirectX 9 features.
- Switching to DirectX 9: Figuring out what's possible and how to do it.
- Shaders for the new millennium:
 Examining the new capabilities of pixel and vertex shaders and figuring out what can be done with the rich, new feature set.

Takeaway: Participants gain a better understanding of the dynamics of cutting-edge graphics hardware, leaving with a cookbook of ideas and high-performance, high-quality rendering techniques applicable to DirectX 9-class graphics hardware. Special attention is given to ensure these ideas scale well across a range of PC hardware.

Sim Dietrich is part of the technical developer relations group at Nvidia, where he assists game developers with graphics optimization and techniques. Sim was a contributor to the book *Game Programming Gems*, and served as a contributor and editor for the graphics display section of *Game Programming Gems 2*.

Richard Huddy has worked in the computer games industry for more than 14 years. For most of that time, he has specialized in high-performance 3D graphics. He worked with both RenderMorphics and Criterion Software developing their low-level APIs before moving on to work as a device driver writer specializing in Direct3D optimization.

105

BUILDING A FLEXIBLE RENDERING SYSTEM FOR HIGH-END CONSOLES AND PCs

Rob Huebner

Rendering articulated 3D meshes is a core component of every 3D game. This indepth tutorial gives attendees the tools needed to create a commercial-quality 3D model renderer for both console and PC platforms. The session begins with the most basic steps of rendering a triangle, progresses through topics such as level of detail, lighting, and skin deformation, and touches upon more advanced topics such as shadow volumes and projection, morphing, and complex shaders. The techniques presented dive directly into the specific 3D APIs; middleware libraries are not used, giving the attendees a view of the full process rather than just the high-level abstractions. Specific code and data structure examples will be included from a variety of console and PC-based platforms.

Takeaway: Attendees are provided with specific examples and code based on Maya, console, and Direct3D rendering, and with information about Nihilistic's own 3D rendering engine for next-generation consoles.

Robert Huebner is a co-founder of Nihilistic Software, a small development team located in Marin County, Calif. Huebner's previous work includes JEDI KNIGHT: DARK FORCES 2 from LucasArts, DESCENT from Parallax Software, and contributions to STARCRAFT by Blizzard Entertainment. Nihilistic's first title, VAMPIRE: THE MASOUERAGE REDEMPTION, was a top-10 seller when released in March 2000. Rob is Nihilistic's director of technology, creating new engines and tools for Nihilistic's next project, which is being developed for next-generation console systems.

106 PROGRAMMING GAMES FOR SHOCKWAVE AND FLASH

Gary Rosenzweig

The world of Shockwave games has changed completely in the last year with the introduction of Shockwave 3D. This tutorial explores how to use the 3D engine to create games that look and feel like big CD-ROM games. Flash game development has also come a long way, and the tutorial shows how to build some simple Flash games and starts attendees on the path to building more complex ones. Participants see numerous examples and get a good overview of the kinds of games these two tools can be used to create.

Gary Rosenzweig runs the world's largest independent Shockwave and Flash game development company. CleverMedia runs four of its own game destination sites and also builds games for other companies. Gary has created more than 100 games with Shockwave and Flash. He also has written several mass-market books on Shockwave and Flash, including Advanced Lingo for Games and Macromedia Flash 5 ActionScript for Fun and Games.

tutorials

107 FASTER MATH FUNCTIONS

Robin Green

This tutorial explains how to efficiently implement complex functions such as arctangent, exponents, logarithms, square roots, and other transcendental functions on modern videogame hardware, with a practical emphasis on vector unit implementation for PlayStation2. This lecture will investigate common floating point "gotchas," range reduction, polynomial approximation, table and semi-table based techniques, ways to leverage SIMD, and a number of other floating-point math tricks. Attendees will leave with a toolbox of techniques for rolling their own functions optimized for speed and accuracy of storage. Other topics addressed include: When can software outperform dedicated hardware? What is the right way to use table lookups? What is the best way to "fake" sqrt()? How can packages like Maple and Mathematica enhance my engine?

Intended audience: Control freaks, speed freaks and size freaks who are looking for hardcore information and the ability to fully understand floating-point math. Programmers on all platforms benefit, including low-level PC library coders, console hackers, handheld jockeys, math junkies, or anyone looking for that extra 10% edge over the competition.

Prerequisite: A working knowledge of floating-point formats, knowledge of assembly and college math. Warning: Equations are shown. Lots of them.

Robin Green has been in the computer games industry for seven years and currently works as an R&D programmer for Sony Computer Entertainment America. Previously he has worked as a technical director at Electronic Arts and as R&D programmer and C++ language police at Bullfrog Productions

RUNNING MASSIVELY MULTIPLAYER GAMES

Raph Koster, Gordon Walton, Rich Vogel

With the tremendous success of massively multiplayer role-playing games such as ULTIMA ONLINE, EVERQUEST, and ASHERON'S CALL, and the arrival of new genres of games such as Motor City Online, Majestic, and The Sims Online, many developers and publishers are jumping headlong into the development of massively multiplayer experiences. Hard as it is to develop a title, it's only half the battle. Once the game is launched, it faces a host of other concerns that challenge traditional design, development, and production paradigms. This tutorial covers the challenges of running the service side of the massively multiplayer equation, focusing primarily on design, production, and service issues.

Among the topics to be covered are:

- Differences between a development team and a "live" team
- Production principles in a live environment
- · Live team make-up
- Minimal-risk development pipelines and processes
- QA issues on large-scale games
- · Designing ongoing content
- What do you add over time?
- Balancing the players' and the development team's needs

- How to set up a service infrastructure
- Handling troublemakers and "grief players"
- Evaluating play patterns and maximizing your service

Takeaway: Participants gain an understanding of the differences between live services and development projects; customer service patterns and paradigms; and development process models for updating games after they are live.

Raph Koster was the lead designer for Origin's ULTIMA ONLINE and ULTIMA ONLINE: THE SECOND AGE. He is creative director at Sony Online Entertainment, where he is heading up the design of STAR WARS ONLINE. He maintains a web site of writings on these topics at www.legendmud.org/raph/gaming.

Rich Vogel was the producer of MERIDIAN 59 and Origin's ULTIMA ONLINE. He is currently executive producer at Sony Online Entertainment, where he is overseeing the development of STAR WARS ONLINE

Gordon "Tyrant" Walton started writing commercial computer games in 1977. He is current vice president and executive producer of THE SIMS ONLINE at Maxis.

109 EFFECTIVE PROJECT MANAGEMENT

Randy Angle & William Dwyer

What happens when an architect and an engineer tackle project management? Using the principles of good product development learned through experience before joining the game industry and the lessons learned by surviving disasters afterward, these two developers were able to create a successful and powerful style of project management. This session provides lessons in step-by-step planning tactics and hands-on project management activities. This session is intended for team leads in various disciplines, including engineers, artists, designers, and producers. If you are responsible for some aspect of a project and would like to succeed without stepping on the team, attend this session.

Randy Angle, a lead engineer and designer for Stormfront Studios, has published several articles and given presentations at industry conventions on game design, programming and project management. He has just finished two Lego titles for Stormfront as designer and lead engineer.

William Dwyer left architecture after 10 years of doing historical preservation, designing corporate interiors and single family homes, he joined Stormfront in 1993 to work on TONY LA RUSSA BASEBALL.

110 BEYOND STORYTELLING

Lee Sheldon

This tutorial goes beyond the myths that story and gameplay don't mix and that by simply creating an environment for players to play in, meaningful human drama somehow emerges. The session draws from literature, drama and film those elements of cross-media storytelling that also apply to games: character, dialogue, conflict, pace, and more. Participants learn practical techniques for creating emotionally involving gamestories in solo computer games, console games and persistent worlds; study games in current release that tell stories; examine the pros and cons of author-written stories and player-written stories; and explore the story keys that can unlock the elusive mysteries of the true mass market and gain us the critical respect for our products enjoyed by other media. Throughout the day, participants play a modular gamestory that can be experienced in any order and adapts to the choices players make without sacrificing gameplay or the power of the story itself.

Takeaway: Participants learn games can contain stories as powerful as those in any other medium and are given some of the tools used to create such games.

Lee Sheldon was trained as a director in theatre and film, and has more than 200 television production credits, from CHARLIE'S ANGELS to STAR TERE: THE NEXT GENERATION. He has been a designer, writer, director, or producer of interactive entertainment for such titles as THE GRYPHON TAPESTRY persistent world, and the Cyan Worlds Inc. upcoming persistent world based on MYST.

111 MAINSTREAM, MASS MARKET AND MOBILE: GAMING ON THE GO

Chris Wright

Recent research has shown that games are already the most popular service for wireless Internet users. With mobile network operators reliant on new and compelling wireless Internet services to recoup the costs of investment in 3G and future network strategies, the opportunities for the games industry are great. Indeed, companies not becoming involved in wireless games today run the risk of being left behind. This tutorial is designed to be a one-stop crash course in wireless games.

This tutorial takes a practical look at all things wireless. This hands-on workshop focuses on the process of game design, implementation, delivery and distribution to a wireless platform.

Highlights include:

- **Design:** What works on the wireless platform? The session features exercises that allow participants to work through from the conception of an idea to the beginnings of a design.
- **Platform:** The inner workings of a wireless entertainment platform.
- The road to standardization: Writing code to wireless APIs.
- Technologies: SMS, WAP, GPRS, Java, and their opportunities and implications on game development.
- Post-Mortems and Case Studies

Chris Wright is a pioneer of wireless entertainment having been actively involved from the industry's conception. Chris is business development director at Digital Bridges and manages the business relationships within the interactive entertainment market, with publishers, developers and licensors, and also heads the internal production studios of wireless games

ONE DAY TUTORIALS DAY TWO - WEDNESDAY, MARCH 20

201

THE RETURN OF 1001 NIGHTMARES: A CHARACTER DESIGN WORKSHOP

lain McCaig

Come make monsters again! Artist lain McCaig (STAR WARS: EPISODE I AND II, INTERVIEW WITH THE VAMPIRE, and HARRY POTTER AND THE SORCERER'S STONE) returns for an all-day workshop revealing the tricks of the trade of a character designer and creating a new universe of creatures live on stage. This workshop discusses some of the visual storytelling behind such films as INTERVIEW WITH THE VAMPIRE, STAR WARS: EPISODE I, and DINOTOPIA, unveiling the invisible yet essential story art behind the screen. Make ideas live and breath during this interactive presentation.

But be warned: this talk is highly interactive and participation is a must. The workshop is extremely hands-on. Paper, pencils, and passion are essential.

lain McCaig is one of the motion picture industry's finest artistic directors and concept designers. He recently created principal character and costume designs for STAR WARS: EPISODE II: THE PHANTOM MENACE, as well as the forthcoming STAR WARS: EPISODE II: ATTACK OF THE CLONES. His design credits include James Cameron's TERNINATOR II, Steven Spielberg's HOOK, Francis Ford Coppola's BRAM STOKER'S DRACULA, Neil Jordan's Interview WITH THE VAMPIRE, and Sony Pictures' DINOTOPIA.

202

MAYA FOR REAL-TIME ARTISTS

Tim Brown

This tutorial focuses primarily on asset creation for real-time applications. The session covers character creation using some of the new tools in Maya 4.5, including using subdivision surfaces as a modeling tool. Texturing is also covered, highlighting improved workflows for manipulating UV information. The session explores building environments with Maya's toolsets as well as creating lightmaps and color per vertex information. Nonlinear animation enhancements, along with FK/IK switching and the advantages of quaternion rotations round out the character setup and animation workflows. Using Paint Effects, attendees are shown how Maya can be used for texture creation.

Tim Brown is a technical editor at Alias|Wavefront. He is a co-author of *The Art of Maya* book and has contributed to the *Learning Maya* book as well. Tim's focus is the games industry and he is currently creating gamesrelated education materials for Alias|Wavefront customers.

203 ADVANCED OPENGL FOR GAME DEVELOPMENT Cass Everitt

Highly programmable GPUs are moving rapidly into mainstream use and they fundamentally change the nature of 3D programming. It is no longer sufficient to simply render textured polygons. Quality shading requires a firm grasp of the advanced features available on today's hardware. This tutorial covers many advanced rendering topics in detail, starting with the mathematical principles behind them. The corresponding OpenGL interfaces are presented and followed by suggested applications for next-generation games. Demos are shown to illustrate the concepts. The techniques to be presented not only increase realism in games, but also increase performance by offloading many operations onto the GPU that were previously performed on the CPU.

Intended audience: This session is aimed at programmers who have solid OpenGL experience and basic knowledge of linear algebra (particularly vector and matrix operations). Since more and more of these features are exposed in the latest digital content creation packages, this course is essential for tools and plug-in writers and those creating cutting-edge effects and content for next-generation games.

Cass Everitt is an engineer in the OpenGL Applications Engineering group at Nvidia. Prior to joining Nvidia, he worked in the scientific visualization branch of computer graphics, focusing on oceanographic and meteorological model visualization.

REHAVIOR REE

AI BEHAVIOR REPRESENTATION TECHNIQUES FOR COMPUTER GAMES

John Laird & Michael van Lent

This tutorial presents the core artificial intelligence techniques for representing knowledge and making tactical decisions. These are the techniques that all game Al programmers should have in their tool boxes for adding intelligence to any and all aspects of computer games. The techniques covered include finite state machines, Bayesian networks, decision trees, rule-based systems, neural networks, and planning. The tutorial provides an in-depth analysis of the operation, implementation, and applicability of these techniques to computer game Al. The techniques are evaluated in terms of their time and space efficiency and the difficulty of using and maintaining them in a game. In addition, the tutorial covers well-estab-

tutorials

ONE DAY TUTORIALS, WEDNESDAY, MARCH 20, CONTINUED

lished AI learning approaches that are applicable to these decision-making techniques. All of these techniques are illustrated using computer game examples.

Takeaway: For each AI technique (finite state machines, Bayesian networks, decision trees, rule-based systems, neural networks, and planning), attendees learn how each technique works and its strengths and weaknesses.

John Laird has been a researcher and educator in Al for the last 20 years. His research is directed at developing human-level Al systems. He is the developer of Soar, a rule-based language that is used worldwide for Al and cognitive modeling research.

Michael van Lent received his Ph.D. from the University of Michigan in 2000. He currently works at the University of Southern California's Institute for Creative Technologies applying AI research and commercial game techniques and technology to the development of immersive training simulations.

205

MATH FOR PROGRAMMERS

Jim Van Verth

As anyone who has visited a local mall lately should know, the majority of games these days are being done in 3D. Programming for these games requires a slightly higher level of math background than before, not only to do graphics but also for such "new growth" areas as simulation and Al. This tutorial provides a toolbox of techniques for programmers interested in improving their 3D background, with references and links for those looking for more information. The focus of the session is to build a simple tank game and show how problems along the way can be solved using 3D mathematical concepts. Topics include basic transformations and model manipulation, camera tracking, simple hierarchical animation, introduction to scene graphs, basic simulation, collision detection and response, and picking. Sample code libraries and examples are provided.

Intended audience: This tutorial is primarily for programmers who are interested in improving their mathematics abilities, or accomplished programmers who want to fill some holes in their background. Attendees are expected to have a freshman-year course in calculus under their belts and an understanding of basic linear algebra concepts such as vectors and matrices. Review notes are provided for those who want to brush up.

Jim Van Verth has been in the game industry for five years, working on many aspects of 3D, including graphics, sound, simulation and Al. He's also spent way too much time in graduate school. Currently, Jim is technical lead on a PlayStation2 project at Red Storm Entertainment.

207

THE BRICKS AND MORTAR OF A DISTRIBUTED

INTERACTIVE VIRTUAL WORLD

Roger Smith

This tutorial presents the most current simulation techniques being developed by the US military to support simulators, simulations, war games, and virtual experiences that train soldiers. These techniques are directly applicable to game programming and other forms of digital entertainment. The programmer-oriented tutorial describes the latest approaches to modeling physical actions such as movement, detection, engagement, and communications, and to representing the land, sea, and air environment in which combat occurs. The tutorial provides the fundamental software architecture to support scalable, networked simulation. It also includes some fundamental principles to which all creators of virtual worlds should adhere.

Takeaway: Attendees learn the fundamental architecture of a distributed simulation, learn about the move-look-comeshoot cycle that dominates most simulations, understand some techniques that are used to optimize performance, understand the core approaches for synchronizing time in a distributed simulation, and hear the principles of modeling as espoused by simulation luminaries.

Roger Smith is an award-winning virtual world developer who has been growing simulation companies and projects for 15 years. He is an international leader in the field of simulation and distributed virtual worlds, serving on the editorial boards of the leading journals and professional conferences and creating the next generation of simulations for the Department of Defense.

208

QUALITY ASSURANCE TOOLS AND TIPS

Evan Birkby

This presentation discusses the practical issues that affect how a game is tested and what tools are needed to make testing efficient and effective. What is needed in a QA database? What types of test plans are effective and what elements should they include? How should bugs be classified? How should testing communicate with development to make the process as effective as possible? How can politics be handled with testing tools? What is the real responsibility of testing in some organizations: is it quality assurance or quality observation? Better understanding of these issues can help anyone involved in the testing process, including leads and managers. Examples of QA database designs and test plans are a part of the presentation. Tools for testing multiplayer games are discussed as well as some differences in testing games on different platforms.

Takeaway: Participants learn about a number of tools used in testing. Traditional testing processes are broken down into simpler components. The why and how of many test procedures is made clear.

Evan Birkby is manager of testing at Totally Games in San Rafael, California. He has developed a number of testing tools, including QA databases, testplans, and milestone documents. Some of the games he has in his testing credits include FALCON 3.0, Top Gun, MASTER OF ORION II, STAR TREK, "A FINAL UNITY" STAR TREK "GENERATIONS," X-WING COLLECTOR SERIES, X-WING ALLIANCE, and currently STAR TREK BRIDGE COMMANDER.

209

DESIGN VISUALIZATION: TOOLS THAT WORK

Paul Schuytema

There are many facets to designing a complete game, and each requires a unique creative brainstorm. This tutorial takes on the exciting task of design visualization by attacking it on two fronts: growing our "creative intelligence" and then applying that unique intelligence to the process of game design. The first part of the tutorial focuses on getting to know our own creative intelligence and the various triggers and tools we can use to nurture and grow our creative minds in the context of game development work. The second part of the tutorial focuses on using a series of visualization tools to brainstorm and explore the various key aspects of game design.

Takeaway: Attendees leave with an active knowledge of how to nourish their own creative growth, plus they are armed with the tools needed to apply their creative mind to specific areas of game design and development visualization.

Paul Schuytema has been involved with games nearly all of his life and with computer games since the late 1970s (with early programming and design work on the TRS-80 Model 1, Apple II, and Atari 800 for publishers like Rocklan and Guinness). He currently heads up the development and design team at Magic Lantern Playware (Tom CLANCY'S RAINBOW SIX COVERT OPS ESSENTIALS, SURVIVOR: THE INTERACTIVE GAME).

210 GAME DESIGN FOR WIRELESS DEVICES

Greg Costikyan

This day-long tutorial examines design and development for wireless devices, particularly Internet-enabled cell-phones, but PDAs as well.

Topics include:

- Technologies, and particularly how they shape the possibilities in game design, including SMS, WAP, J2ME, BREW, messaging integration, location-based services, et al.
- Designing for a highly-connected but media-poor technology.
- The importance of community & player interaction.
- Limitations of the medium, including high latency, form factor, interface considerations, et al.
- Strengths of the medium, including the ability to play any time and anywhere, networked connectivity, and the inherently social nature of wireless devices.
- Specific case studies looking at successful wireless games currently in deployment.

Greg Costikyan has designed 28 commercially produced board, role-playing, PC, online, and wireless games. He is co-founder and chief design officer of Unplugged Games, a wireless games company. He writes frequently about games, game design, and the game industry for publications including *Wall Street Journal Interactive*, *Salon*, and *Game Developer* magazine.

211 DEVELOPMENT CONTRACTS BOOT CAMP

Jim Charne, Dan Offner & David Rosenbaum

The session opens with a brief discussion of the best way to approach a publisher and cultivate interest in a project and a deal. Once the two sides determine they want to move forward, the panel begins to address the issues that must be resolved before a contract can be signed. The panel reviews business models used by developers and publishers in console, computer, handheld, and Internet game development to determine the expectations of each party going into the negotiation. Contract issues to be covered include: ownership of work product, designs, and technology; penalties versus incentives for on-time completion; the publisher and third-party approval process; step-type contracts; change order procedure; the role of firstparty hardware companies in the process; termination for convenience and developer's rights; right to develop or receive payments for sequels, localization, and use of underlying technologies; royalties, recoupment, accounting, royalty reserves,

audit rights, and more. The program closes with a mock negotiation in which concepts discussed during the day are applied.

Takeaway: Participants leave with an understanding of core issues in negotiating development deals that can affect growth, profitability, and continuing viability of a developer as a business.

Jim Charne has represented clients in all facets of interactive software entertainment since the mid-1980s. He entered the industry in 1983 as a producer of Atari 2600 games for Activision. He is admitted to practice law in California, New York, and New Jersey.

David Rosenbaum is a member of Fischbach, Perlstein & Lieberman. Among his numerous interactive clients are Acclaim Entertainment, as well as game developers Surreal Software and Sucker Punch Productions.

212 USING DIRECTMUSIC FOR MUSIC AND SOUND FX

DirectMusic provides a high level interface for creating dynamic, interactive, and non-repetitive sound effects and music. In this tutorial we explore the features this library brings to DirectX and Xbox development from the perspective of composers and sound designers. We cover DirectMusic terminology, concepts, and how to apply them to real-world audio scenarios. Basic and more complex scenarios are authored in step-by-step tutorials using DirectMusic Producer, the authoring tool for creating DirectMusic content.

Topics include:

Scott Selfon

- Basic DirectMusic Concepts
- Using DirectMusic for real-time sound effects creation and manipulation
- Variable content and DirectMusic
- Using DirectMusic to combine MIDI and pre-rendered wave audio
- Using DirectX audio scripting for dialog
- Authoring, organizing, and delivering content using DirectMusic Producer
- Content-driven and self-modifying audio implementations
- Integration between programmer and content creator for DirectMusic
- DirectMusic on the Xbox versus DirectMusic on the PC
- Demonstrations and discussion of titles that make use of varying aspects of DirectMusic

Intended audience: This tutorial is intended for audio content creators.

Scott Selfon is the audio content consultant on the content and design team in the Xbox Advanced Technology Group at Microsoft. He has composed for a wide range of media, including film and television, PC titles, and live

206

Sponsored Tutorial MICROSOFT DIRECTX DAY



DirectX Team

Acquire bleeding-edge knowledge on topics like Microsoft DirectX graphics performance optimizations, shader optimization, DirectX high-level shading, and even material surface physics for shader writers. Get smart in effects in Direct3D, grasp the DirectPlay networking layer in depth and absorb the latest on D3DX and art pipeline support.

Topics include:

- The New 3D graphic accelerator generation of 2002
- Effects in Direct3D
- D3DX and art pipeline support
- DirectX graphics performance optimizations
- Material surface physics for shader writers
- Shader optimization with DirectX 8
- DirectPlay networking layer in depth
- New opportunities: gaming and .NET
- DirectX high-level shading

3

sessions

advance the state of the art

Last year's emerging technologies and gameplay innovations are this year's standards. GDC lectures, panels and roundtables cover the rapidly evolving world of game development with information and inspiration in game design, production, audio, visual arts, business and legal, production and level design across all genres and all platforms. GDC Sessions run March 21-23, 2002.

visual arts

Attendees come away from the Visual Arts track with a better understanding of how to incorporate new techniques in modeling, texturing, design and animation and how to keep up with the exponential art demands of today's games.

Art Direction: The Employee, the Critique, and the Axe

Steve Reid, Red Storm Entertainment

Art Director/Lead Artist Roundtable

Seth Spaulding, Cyberlore Studios

Art Directors Panel

Cyrus Lum, Inevitable Entertainment Dale Mauk, Straham Mauk Studio

Character Animation for Videogames

Ron Friedman, Creative Capers Entertainment

Cinematic Lighting for 3D Graphics

Jeremy Birn, render3d.com

Cloth Simulation: FROM FINAL FANTASY: THE SPIRITS WITHIN to the PS2

Gerard Banel, SquareUSA

Creating 3D Assets that Perform Well on 3D Hardware

Jani Penttinen, Westwood Studios

FINAL FANTASY: THE SPIRITS WITHIN from Movie to Real Time

Kaveh Kardan, SquareUSA

Getting Cinematic Quality from In-Game Cutscenes Without a Hollywood Budget

Hugh Hancock, Strange Company

Hardcore Character Modeling and Animation Paul Steed

Inverse Kinematics: Setup and Techniques

Martin Coven, Inevitable Entertainment

The Making of SHREK

Nick Walker, PDI/Dreamworks

Marrying AI to Animation: The Bright Future of **Animation in Games**

Sunil Thankamushy, Electronic Arts

Modeling Lush and Expansive **External Environments**

Matt Dixon & Ian Lovett, Big Blue Box

Modern Techniques in Creating Tile-Based **Graphics for GameBoy Advance**

Karthik Bala, Vicarious Visions

The Necessity of Character Design in **Game Development**

Bob Rafaei, Naughty Dog

The Power of the High Pass Filter

Peter Hajba, Remedy Entertainment

The Quest for Pure Motion Capture

Daivd Washburn, Westwood Studios

Scripting for Artists: Beyond the Basics

John Versluis, Inevitable Entertainment

Supporting Scalability as Artists

Christian Bradley & Chris Seitz, Nvidia

Too Many Polygons! Artistic Alternatives for Harnessing Hardware

Walter Park, Saffire

Toss it in the Blender: Nonlinear Animation for Games

Steve Theodore, Valve Software

UI Case Study: This is FOOTBALL 2002

Andrew Hamilton & Dave Ranyard, Sony Computer Entertainment Europe

Gamasutra.com

Gamasutra.com delivers daily news, technical articles, streamed video lectures, discussion boards, a comprehensive buyers guide of game development products and services, and one of the largest game development job boards on the web with a new resume posting/search service to over 110,000 game developers worldwide.

www.gamasutra.com

Go to www-gdconf-com for full session descriptions, schedules, speaker bios and the latest updates.

game design

The Game Design track focuses on the skills crucial to mastering the design process for modern games. Designers learn to effectively create and communicate vision and goals to team members, publishers, press and the public, and how to identify and focus on the key aspects of the design process.

20 Secrets for Creating Realistic Dialogue and Scene Flow

David Freeman

The Addictiveness of Games

Steve Meretzky, WorldWinner.com

Beyond Psychological Theory: Getting Data that Improves Games

Bill Fulton, Microsoft

Building a Third-Generation Online Persistent World Game

Raph Koster, Verant Interactive Rich Lawrence, Sony Online Entertainment Jessica Mulligan, The Themis Group Gordon Walton, Maxis

Business and Creative Aspects of Sequels

Warren Spector & Harvey Smith, Ion Storm Austin

Children's Software: Past, Present, and Future

Ken Kahn, Animated Programs

Complexity Demons: They're All Around Us

Andrew Leker, Mind Control Software

Content Patterns in Game Design

Bernd Kreimeier, Oddworld Inhabitants

Creating Characters with Dimension and Depth

Bob Bates, Legend Entertainment

Creating Emotional Involvement in Interactive Entertainment

Mark Barrett, Prairie Arts

Design Fundamentals of Stealth Gameplay in the THIEF Series

Randy Smith, Ion Storm Austin

Disney's Toontown Online: Lessons Learned

Jesse Schell & Joe Schochet, Walt Disney Imagineering

Dynamic Difficulty Adjustment

Dan Arey & Evan Wells, Naughty Dog

Episodic Content: Here, Now, and Next Month

Chris Foster, Turbine Entertainment Software

Episodic Entertainment: Viva La Evolution!

Neil Young, Electronic Arts

The Evolution of the Fantasy Role-Playing Genre

Using PROJECT Ego as an Example

Peter Molyneux, Lionhead Studios

Experimental Gameplay Workshop (see page 18)

Jonathan Blow, Bolt Action Software

Game Designers and Development Teams

David Wessman, Stormfront Studios

The Game Design of VIB RIBBON 2

Masaya Matsuura, creator, PaRappa the Rapper & Um Jammer Lammy

Games with Collectible Components

Richard Garfield, Wizards of the Coast

The Hook(s) of the Game

David Perry, Shiny Entertainment

Inventing New Concepts and Game Mechanisms

Pascal Luban, The Game Design Studio

Laws of Making Massively Multiplayer Games

Anthony Castoro, Verant Interactive

Massively Multiplayer Gameplay System Implementation

Justin Quimby, Turbine Entertainment Software

More of the 400 Rules of Game Design

Noah Falstein, The Inspiracy Hal Barwood, LucasArts Entertainment

Multi-Scale Community Design for Games

Amy Jo Kim, NAIMA

Not a Product: Design Similarities Between Massively Multiplayer and WAP Games

Adam Mayes, Digital Bridges

Online Anarchy: Massively Multiplayer from a Design and Architectural Point of View

Tommy Strand, Funcom

Online Game Design for Mobile Phones

Ryo Shimizu, DWANGO

PC-Enabled Toys: Are You a Game Designer or Inventor?

Robert Nashak, Vivendi Universal Interactive

Practical Game Analysis with Doug and Warren

Doug Church & Warren Spector, Ion Storm Austin

Pros and Cons of the Organic Design Process

Dene Carter & Simon Carter, Big Blue Box

Real-Time Strategy Game Balance

Dustin Browder, Westwood Studios

REZ: The Synesthesia that Games Invite

Tetsuya Mizuguchi, creator, SPACE CHANNEL 5 & SEGA RALLY CHAMPIONSHIP

SEAMAN: A Case for Making Eccentric Titles

Yoot Saito, Vivarium

The Secret of Psalm 46

Brian Moriarty, Skotos Tech

Stories You Can't Tell

Chris Crawford

Story and Gameplay are One: The Grand

Unified Theory of Entertainment

Jesse Schell & Joe Shochet, Walt Disney Imagineering

Storytelling in the Online Medium

Rich Vogel, Verant Interactive

This Is Not a Game: A Discussion of the Creation of the Al Web Experience

Elan Lee & Jordan Weisman, Microsoft

UI Design Roundtable

Garner Halloran, Red Storm Entertainment

Web-Based Games Roundtable

Brian Robbins, CleverMedia

What Worked: 10 Detailed Examples of Successful Character Action Design

Mark Cerny, Cerny Games

When Maps Collide: A Conversation with

Will Wright and Scott McCloud

Why We Shouldn't Make Games

Ernest Adams

Wireless Game Design Review

David Collier, PacketVideo

production

The Production track passes along the hard-earned knowledge and practices that drive the industry's most consistent and successful teams. The track focuses on providing the skills and knowledge necessary to protect and nurture game development teams and their projects.

A Method to the Madness: Preproduction Methodology Chris Ulm, Oddworld Inhabitants

Building a Better Bug Trap

Michael Weiner

Creating Useful Technical Design Documents

Steve Taylor, Wahoo Studios

Effective Team Communication

Heather Kelley, Sapient Austin

How to Manage a Large-Scale Online Gaming Community

Rich Vogel, Verant Interactive

Know Your Audience: Writing Design Documents that Get Read

Brian Upton, Red Storm Entertainment

Managing Large Volume Text Translations in Computer Games

Don Moar & Chris Christou, BioWare

Managing the QA Process

Chuck McFadden, LucasArts Entertainment

Preproduction

Warren Spector, Ion Storm Austin

Production within a Large Publisher

Tony Goodman, Ensemble Studios

Quality or Death: How the Business Model and Player Psychology Collide in For-Pay Online Games

Eric Todd, Maxis

Realistic Expectations in Developer/Publisher Relations

Haden Blackman, LucasArts Entertainment Tom Frisina, Electronic Arts Tony Goodman, Ensemble Studios Geoff Keighley, GameSlice.com Stuart Moulder, Microsoft PC Games Brian Raffel, Raven Software Jay Wilbur, Epic Games

Real Methods of Game Production

Erik Bethke, Taldren

Redefining the Wheel

Judith Lucero

The Role of Technical Director at LucasArts

Scot Brew, LucasArts Entertainment

Scalability: The Key to Successful PC Development Sanford Russell, Nvidia

Staying on Time and on Budget: Processes, Milestones, and Schedules

Don Daglow, Stormfront Studios

Tracking Assets in the Production of Final Fantasy: The Spirits Within

Shiro Kawai, SquareUSA

Two Dozen Ways to Screw Up a Perfectly Good ProjectDave Rohrl, pogo.com

level design

The Level Design track focuses on real-world design problems and solutions discovered in the development of hit games. Veteran level designers from various genres share their experience and lessons learned on successful titles.

Battling Level Design in Hardcore Genres for a Casual Audience

Tim Longo, LucasArts Entertainment

Creating More from the Less of Level Design Ivan Beram

Digital Imagery and Level Design

Aki Maatta, Remedy Entertainment

From All Levels: Perspectives of Level Design

Tim Miller, LucasArts Entertainment

Game Editing Tools

David Weinstein, Red Storm Entertainment

Genre Specific World Building Methodology in SSX & SSX TRICKY

Sinisa Karolic, Electronic Arts, Canada

Intelligent Level Design

Mark Meier, Art Institutes International

Level Design in an Outdoor Environment

Jolyon Leonard, Innerloop Studios

Manhattan as Muse: New York City as a Conceptual Tool

Duncan Brown, ZenLux

May Time Be with You: Level Designing ROGUE LEADER

Chris Klie & Albert Chen, Factor 5

Progressive Level Design: Systems-Based Level Design for Emergent Gameplay

Harvey Smith, Ion Storm Austin

Team Design Tactics and Approaches to Photorealistic Styles of Level Design

Benson Russell, 2015

Technology's Impact on Level Design

Scott Blinn, Volition

The Tricks and Traps of RTS Campaign, Scenario and Deathmatch Map Design

Ian Kilmon, Timegate Studios

Register by February 5, 2002 to save up to 30%

Check www-gdconf-com for the latest track updates.

programming

The Programming track brings together top coders from around the world to share their latest ideas. Case studies help programmers of all levels gain valuable real-world experience, while more academic courses introduce concepts certain to be next year's features.

3D Performance and Optimizations for the ARM Architecture Haim Barad, Intel

Action-Based Discretization for Al Search

Todd Neller, Gettysburg College

A Data-Driven Game Object System

Scott Bilas, Gas Powered Games

Al in Computer Games Roundtables and Interactive Discussion

Eric Dybsand, Glacier Edge Technology Steven Woodcock, Neil Kirby

Al in Strategy Games

Denis Papp, Timegate Studios

Another Five Years from Now: Future Technologies

David Braben, Frontier Developments

Building a Data-Driven Game Engine: A Case Study from AGE OF MYTHOLOGY

Robert Fermier, Ensemble Studios

By the Books: Solid Software Engineering for Games

Brian Sharp, Ion Storm Austin

Clipping on the PlayStation2

Colin Hughes, Sony Computer Entertainment Europe

Defense Against Service Theft, Internet Cheating, and **Denial of Service**

James Gray, RSA Security

Distributed Area Lighting

Chas Boyd, Microsoft

Fundamentals of Lighting and Perception: The Rendering of Physically Accurate Images

Philip Dutre, Cornell University Hector Yee, Westwood Studios

Fur Using Shells and Fins

Jerome E. Lengyel, Microsoft Research

Game Boy Advance Networking

Cathryn Mataga, Junglevision Software

Game Scripting in Python

Bruce Dawson, Humongous Entertainment

Hardware-Accelerated Procedural Texture Animation

Greg James, Nvidia

High Dynamic Range Imagery and Image-Based Lighting Paul Debevec, USC Institute for Creative Technologies

Image-Based Rendering for Computer Games

Radek Grzeszczuk, Intel Microprocessor Research Labs

Implementation of Group Minds (Lebensform)

Richard Evans & Tom Barnet Lamb, Lionhead Studios

Incredibly Dense Meshes

David Brickhill, Activision

In-Depth 3D Exporter Design and Implementation

Casey Muratori, RAD Tools

In-Game Special Effects and Lighting

Tomas Arce & Matthias Wloka, Inevitable Entertainment

Life as a Lead Programmer

David Weinstein, Red Storm Entertainment

Level-of-Detail Rigid-Body Physics

James Golding, MathEngine

Making It Easier to Develop for Consoles Roundtable

Mark Thomas, Microsoft Xbox

Maximizing DMA Bandwidth on the PlayStation2

Greg Omi, Naughty Dog

Multiplayer Tricks of the Trade

David Weinstein, Red Storm Entertainment

Of Internet Servers and SQL Databases: Designing the Backend for Power and Performance

Peter Hallenberg, Incredible Technologies

Optimizing Memory Bandwidth

Mike Wall, Advanced Micro Devices

Polygon Soup for the Programmer's Soul: 3D Pathfinding

Patrick Smith, Westwood Studios

Realistic Deformation

James O'Brien, University of California, Berkeley

Real-Time Cloud Rendering for Games

Mark Harris, University of North Carolina at Chapel Hill

Real-Time Hatching

Cem Cebenoyan, Sebastien Domine & Ashu Rege, Nyidia

Rendering Outdoor Light Scattering in Real Time

Nathaniel Hoffman, Westwood Studios

Scripting Languages Roundtable

James Lane, Mare Crisium

Shader Integration: Merging Shading Technologies on the Gamecube

Florian Sauer & Sigmund Vik, Factor 5

Sound Propagation in 3D Environments

Chris Carollo, Ion Storm Austin

Software Engineering in the Game Industry Roundtable

Noel Llopis, Meyer/Glass Interactive

Taming a Wild River: 3D Fluid Simulation

Jeff Lander, Darwin 3D

Using a Webcam as a Game Controller

Jonathan Blow, Bolt Action Software

Vector Units and Quaternions

Jim Van Verth, Red Storm Entertainment

The Visual Engineering of SSX: A Post-Mortem

Mike Rayner, Electronic Arts, Canada

Game Developer magazine

The first print publication written specifically for creators of entertainment software, Game Developer magazine provides technical and industry information to over 35,000 professional game developers. Count on Game Developer magazine for the most relevant and respected content in the game industry.

www.gdmag.com

business and legal

The Business and Legal track sessions provide critical advice for growing and protecting a game development business and help attendees from all backgrounds improve the operations, management, legal and business aspects of their companies.

Beyond VC: Alternative Financing Models for Game Development

Jason Kay & James O. Thoma, Akin, Gump, Strauss, Hauer & Feld

Big Huge Games: A Year (or Two) in the Life of a Startup

Brian Reynolds & Tim Train, Big Huge Games

Company Culture: Why You Need It, How You Get It Kathy Schoback, Sega of America

Console and PC Game Trends

Howard Dyckovsky & Steve Koenig, NPDTechworld, Ilene Haase & Richard Ow, NPDFunworld

Developing in Developing Countries

Glyn Anderson, Matahari Studios

The Employment Life Cycle: Your Best Investment Robin Longoria, CMP Media

Enter the Dragon: The Chinese Game Industry
Zhan Ye. Game Software, China

Finding a Niche in a Crowded Marketplace

Phil Steinmeyer, PopTop Software

Following the Money

Louis Castle, Westwood Studios

The Freelancer's Roundtable

François Dominic Laramee

Global IP Protection and Anti-Piracy Techniques Chris Chapin, Electronic Arts

How to Get Positive Coverage for Your Company

Sue Bohle, The Bohle Company How to Make a Million Dollars On the Web:

A Case StudyAustin Meyer, Laminar Research

IGDA White Paper on Online Games

Alex Jarett, Internet Executives Club

Innovation on Demand

Kevin Bachus

Intellectual Property: The Current Game of Swords and Shields

Stephen Rubin, esq.

Interviewing Techniques

Dan Baker, Saffire Corporation

Mobile Game Business Models

David Collier, PacketVideo

Mobile Phone Games: Where Are They Now and Where Are They Going?

Ian Baverstock, Kuju Entertainment

Moving to Multiple Projects

Dr. Ray Muzyka & Dr. Greg Zeschuk, BioWare

Negotiating Contracts: Getting Developers What They Need While Giving Publishers What They Want

Chuck Camps, RKG Camps

Negotiating the Best Deal Possible

Jay Powell, Octagon Entertainment

Negotiating with Publishers: Issues, Rights, and Strategies

Joshua Grode, Liner & Yankelevitz

Patents: Are Things Getting Better?

Tim Wu, Riverstone Networks

Publishers Speak 2002: What it Really Takes to Get a Publishing Deal

Dan Rogers, BizDev

Software Contracting as an Xtreme Sport

Jim Charne, Law Offices of James Charne

Startup Horror Stories

Tim Morten, Savage Entertainment

Venture Capital Bootcamp for Game Startups

Mark Long, Zombie

Who Owns What? Software Patents and Intellectual Property in Games

Tobi Saulnier, Vicarious Visions

IGDA

The IGDA's mission is to build a community of game developers which leverages the expertise of our members for the betterment of the industry and the development of the art form. To achieve this, we need your support, and there's no better time to get involved than now, since your membership earns you \$50 off the registration price of the GDC. Get a membership application and full details at:

www.igda.org or join when you register for GDC (see registration form).

Check www-gdconf-com for the latest descriptions and updates.

audio

Attendees come away from the Audio track with a better understanding of the current state of interactive game audio. Sessions cover composition, sound design, alternative platforms and the project management necessary to make games sound better and deliver on time.

Adapting Cinematic Film Scores for Games

Rod Abernethy & Dave Adams, Slackmates

Afterthoughts: The Audio of Rogue LEADER

Thomas Engel, Factor 5

Audio Business Issues

Rich Goldman, RipTide Music

Audio Production for HALO

Marty O'Donnell, Bungie Studios

Console Audio Panel

Brian Schmidt, Microsoft, Clint Bajakian, C.B. Studios Alex Brandon, Ion Storm Austin, Rob Hubbard, Electronic Arts Thomas Engel, Factor 5

The Complete 5.1 Audio Experience for Videogames

Murray Allen, Electronic Arts & Tommy Tallarico, Tallarico Studios

Dolby Pro Logic II: Here's the Deal - Sponsored Session

Jack Buser & Kristoffer Larson, Dolby Developer Support Engineers

Dynamic Range in Games: An Overview of Sound Creators

Tom Hays, NovaLogic

Game Audio Production: Process and Strategy

Clint Bajakian, C.B. Studios

Game Composers Union Roundtable

Tommy Tallarico, Tallarico Studios

Interactive Composition Roundtable

Alexander Brandon, Ion Storm Austin

Maintaining an Edge in this Increasingly Competitive Field

Brian Schmidt, Microsoft

Managing Audio Development on Large Projects

Heather Sowards, VR1 Entertainment

Practical Tips and Tricks in Sound Design and Recording

Iris Roane, StarSapphire Studio

Using a Live Orchestra in Game Soundtracks

Matt Uelmen, Blizzard, Chance Thomas, HUGEsound Network Tommy Tallarico, Jeremy Soule, Michael Giacchino

Web Audio Roundtable

Steve Horowitz, Nickelodeon Online

Xbox Audio

Scott Selfon, Microsoft

IGDA Roundtables & Discussion Groups



A Beginner's Guide: The Building of an Industry Professional

John Feil, LucasArts Entertainment
David Weinstein. Red Storm Entertainment

Academia and Videogames: How to Build & Maintain a Relationship Between Two Different Worlds

John Buchanan, Electronic Arts, Canada

Are Massively Multiplayer Games Blazing a New Trail for Humanity?

Patricia Pizer

Character Artists Discussion Group

Jason Wiener

Cultivating a Games Industry in Developing Countries

Gino Yu, Hong Kong Polytechnic University

The Cultural Study of Games: More than Just Games

Matthew Southern, International Centre for Digital Content

Developer Activism

Jason Kingsley, Rebellion, Jason Della Rocca, IGDA

Developer Quality of Life

François Dominic Laramee

Director, Shockwave, and Flash Game Developers Discussion Group

Brian Robbins, CleverMedia

Ex-Demosceners Discussion Group

Paul Bragiel, Paragon Five

Game Programming Gems Discussion Group

Mark DeLoura, Founding Editor, Game Programming Gems

Getting Students Involved: Working on Establishing IGDA Student Chapters

Jeff Ward, James Madison University

How to Organize and Run an IGDA Chapter

Steve Meretzky, WorldWinner.com

IGDA Chapter-Building Strategies

Jason Della Rocca, IGDA

Intersections: Using Games Outside Entertainment

Anders Frank, Swedish Defense Game Studio

Learning from the Classics: A Discussion of What Made Games Great

Alex Neuse, LucasArts Entertainment

Making Learning and Training Games That Don't Suck

Marc Prensky, games2train.com

Minorities in Game Development

Darrell Porcher, Sony E-Solutions

Rookie Studios: Game Design Resource Management

Jim McNally, Longbow Digital Arts

Rookie Studios: Fantasy Island or The Real World?

Kent Quirk, CogniToy

Rookie Studios: Fun in the Long Run

Kirk Owen, Octagon Entertainment

The Phenomenology of Game Design

Thomas Buscaglia, BallroomGames

Secrets of Successful Indie Developers

Steve Pavlina, Dexterity Software

Studio Affiliate Series: Making Money. Period.

Louis Castle, Westwood Studios

Studio Affiliate Series: Developer Negotiation Tactics & Strategies

Dan Rogers, BizDev, Students Groupie Jeff Ward, James Madison University

Teachers and Other Academics: A Discussion Group

Celia Pearce, University of California, Irvine

Toward Standardization of Al Interfaces

Alexander Nareyek, GMD - FIRST

Violence: New Perspectives on Old Designs

Daniel Greenberg

Who Are You Kidding? Play Testing for Developers

Michael Meischeid & Tobi Saulnier, Vicarious Visions

Women's Careers in Game Development

Ellen Beeman, Beemania Software Design & Production Consulting

Women in Game Development: On Your Mark, Get Set...

Sheri Graner Ray, Metrowerks

events at GDC

game developers choice awards

Celebrate the developers of 2001's best games. The Game Developers Choice Awards are where the game industry pays tribute to the art, effort and sacrifices that go into making truly innovative interactive entertainment.





4th annual independent games festival

Experience unsigned games that break genres, push the boundaries of gameplay and redefine the interactive experience. Play the finalist games at the IGF Pavilion, and see this year's winners at the Game Developers Choice Awards ceremony. **www.igf.com**





Craft Sponsor:

experimental gameplay workshop

The Experimental Gameplay Workshop is a gathering of game developers interested in new and risky game designs. Presentations by experimental game authors is followed by peer discussion. The goal is to provide a platform for game designers to showcase risky new work and discuss it with their peers, legitimize gameplay research and development, and create a community of experimental game designers.

For more information or to present your game, email jon@bolt-action.com.

figure drawing workshop

Taught by Juan Ortiz

game room

Relax with good old low-tech games like Go, Settlers of CATAN, POKER and RISK.

real-time reel

See the year's best real-time game art clips. The GDC Real-Time Reel celebrates the importance of real-time gameplay and its contribution to a game's overall design. Enter your clip by February 1, 2002.

GameDevelopers
Conference real-time reel

www.gdconf.com/reel

suite night

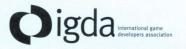
Network with other industry professionals while enjoying food, drink and entertainment hosted by GDC exhibitors.

the programmer's challenge 2.0

The GDC's favorite game show is returning in 2002. Watch your favorite programming stars answer brutally sarcastic questions about all things gaming as they square off in a hilarious duel between the west coast of the US and the rest of the world.

IGDA annual meeting

The IGDA is the independent, non-profit professional association for developers of interactive entertainment. All are welcome at the IGDA's Annual Meeting. Come discover how you can help make this industry the best it can be.



GDC 2002 schedule

Tuesday March 19	Wednesday March 20	Thursday March 21	Friday March 22	Saturday March 23
Registration open 7am-4pm	8am-4pm	7:30am-7:30pm	8am-7:30pm	8am-4pm
		GDC Conference Sessions 9am-6:30pm	9am-6:30pm	9am-6:30pm
		Expo Suites Open 9am-7:30pm	9am-7:30pm	9am-4pm
Tutorials 10am-6pm				Wireless Game Summit
IGDA Academic Sum	ı mit 10am-6pm			9am-6:30pm
		Expo Floor Open 11:30am-7:30pm	11:30am-7:30pm	11:30am-4pm
		Independent Games	Festival Runs continuously, Thur	and out the seconds Control of
unch 1-2:30pm			Runs continuousty, Thur	sday through Saturday
			IGDA Annual Meeting 1:30-3pm	
		Real-Time Reel Runs		through Saturday
		Experimental Gameplay Workshop 5-7pm		
			Booth Crawl & Programmers Challenge 2.0 6:30-7:30pm	
	Game Room Open \	Wednesday through Saturday,	Programmers Challenge 2.0 6:30-7:30pm	
	Game Room Open \	Wednesday through Saturday, Game Developers Choice Awards 7:30-9pm	Programmers Challenge 2.0 6:30-7:30pm	

advisory board



Hal Barwood LucasArts Entertainment



Rob Huebner Nihilistic Software



Louis CastleWestwood Studios



Cyrus Lum Inevitable Entertainment



Doug Church Ion Storm Austin



Masaya Matsuura NanaOn-sha



Mark DeLoura Founding Editor, Game Programming Gems



Dale Mauk Hypnos Entertainment



Alex Dunne Gamasutra.com



David Perry Shiny Entertainment



Julian Eggebrecht Factor 5



Jason Rubin Naughty Dog



Chris Hecker definition six



Jez San Argonaut Software



Elaine Hodgson Incredible Technologies



Alan Yu Game Developers Conference

Audio Advisory Board

Buzz Burrowes Sony Computer Entertainment

Brian Schmidt Microsoft

Tommy Tallarico Tommy Tallarico Studios

IGDA Track Chair

Jason Della Rocca International Game Developers Association

pass options

Register online at **www.gdconf.com** to get an additional **\$25 discount** off any package when you use your priority code on the mailing label.

Pass Dates	Features	Early Registration	On or after February 6, 2002
T W TH F S	Access to any of the over 300 sessions, roundtables & keynotes GDC Expo Breakfast & lunch Proceedings All events & parties	\$825 save \$470	\$1295
T W TH F S	One two day tutorial or one tutorial on each day on March 19-20, or IGDA Academic Summit Access to any of the over 300 sessions, roundtables & keynotes GDC Expo Breakfast & lunch Proceedings All events & parties	\$1225 save \$470	\$1695
	All Giga Pass features Wireless Game Summit	\$1525 save \$425	\$1950
Wireless Plus Pass T W TH F S O O ● ● ●	All Classic Pass features Wireless Game Summit	\$1125 save \$470	\$1595
Wireless Pass T W TH F S O O ⊕ ⊕ ●	All Expo Pass features Wireless Game Summit	\$450 save \$100	\$550
T W TH F S	 Audio tutorial Sessions & Keynote for the Audio track only GDC Expo Breakfast & lunch Proceedings All events & parties 	\$525 save \$100	\$625
Audio Pass T W TH F S O O • • •	Sessions & Keynote for the Audio track only GDC Expo Breakfast & lunch All events & parties	\$425 save \$100	\$525
Tutorials Only Pass T W TH F S ■	 One tutorial on each day on March 19-20, or IGDA Academic Summit GDC Expo Breakfast & lunch on Tuesday & Wednesday All events & parties 	\$450 save \$100	\$550
Expo Pass T W TH F S O O • • •	 GDC Expo Five introductory classes (see www.gdconf.com for details) Limited events: Booth crawl, Exhibitor suite night only. 	\$150 save \$45	\$195

travel

The GDC Housing Department offers discounted hotel rates to all attendees. To book your hotel room online, go to **www.gdconf.com/services** or call 866-477-5444 (toll-free) or 415-947-6661 (international/local). Discounted rates are based on availability and may not be available after February 18, 2002.

GDC Hotels:

Headquarters Hotel: Fairmont Hotel

- San Jose Hilton
- Crowne Plaza San Jose
- Hyatt St. Claire
- Hyatt San Jose Airport
 San Jose Convention Inn
- Wyndham Hotel

Car Rental: AVIS is the preferred rental car agency for the GDC. Call AVIS directly at 1-800-331-1600 and give the meeting code: J867631 for discounted rental rates.

Airlines: Discounted air fares are offered to you by our preferred airline, United Airlines. Call to book today at: 1-800-521-4041 and use the code: 502BV.

Avoid the line, register on time.

Advance registration for the GDC closes on Tuesday, March 12, 2002 at 4pm pacific time. Registrations after this date must be submitted on-site at the San Jose Convention Center.

STEP 1 CHO				
For detailed session information, see our website at				
www.gdconf.c	om. Please c	heck your	choices be	elow.
☐ Classic Pas	s • Save \$47	1		
By 2/5 \$825			aach	
	On or arter 2	./0 \$1273	eacii	
\$				
☐ VIP Pass • §	Save \$425			
By 2/5 \$1525	On or after	2/6 \$195	0 each	
\$				
☐ Giga Pass •	Save \$470*			
By 2/5 \$1225		ο// Φ1/ ΩΕ		
	On or arter .	2/0 D1070	each	
\$				
■ Wireless Pl	us Pass • Sa	re \$470		
By 2/5 \$1125	On or after	2/6 \$159	5 each	
\$				
☐ Audio Plus	Pace • Savo 9	100		
By 2/5 \$525			ach	
	On or arter 2	/O DOZJE	acii	
\$				
☐ Tutorials 0:				
By 2/5 \$450	On or after 2	/6 \$550 e	ach	
\$				
■ Wireless Pa	ss • Save \$1	00		
By 2/5 \$450			ach	
\$	on or arter 2	/ O \$000 C	acii	
■ Audio Pass				
By 2/5 \$425	On or after 2	/6 \$525 e	ach	
\$				
☐ Expo Pass •	Save \$45			
By 2/5 \$150		/6 \$195 e	ach	
\$				
*OULOGE THE	001416		0: 5	
*CHOOSE TUT				
you must choo				
Tutorials Only			hoose one	Tutorial
each day. (<i>Circle</i>				
	orials, Marc	h 19-20:		
150 151				
	orials, Tueso			101
101 102	103	104	105	106
107 108		110	111	
	orials, Wedn			
201 202	203	204	205	206
207 208	209	210	211	212
IGDA MEMBER				
Members of the				
package. Disco	unt does NO1	apply to	Expo Pass.	
☐ I am a meml	per of the IGE	A, or lam	signing u	p below,
	acting my \$50			
\$				
Current men	nber#			
		_(required		
Forgot your membership	number? Call the IGI	A Membership	Services Desk at	415.947.6215.
The IGDA Membership D	esk is iiut retated to t	ie negistration l	repartment	
☐ Yes, I would	like to join t	he IGDA fo	or 1 year.	

STEP 2 CHECK METHOD OF PAYMENT

Payment in FULL by check or money order (drawn in US funds) or credit card must accompany your registration in order to be processed. Sorry, no purchase orders. You are not registered until you receive confirmation online, by fax or in the mail from the GDC. To be eligible for discounts,

we must receive your registration by the date specified in the discount promotion – postmarks are not accepted. Make all checks or money orders payable to

Game Developers Conference.

FIIOIIE	Fax
Email*	-
Country/Postal Code	
City/State/Zip	
Mailstop	
Address	
Company	
Professional Title	
Name	
STEP 3 This information is required in orde	er to process your registration.
Signature Federal Tax ID # 11-2240940	
Name on Card	
Expiration Date	
Card #	
□ Charge my credit card: □ Visa □ Mastercar	rd 🗆 American Express
Check #	

How to register

Product and service offerings from other CMP Media properties.

that we may send to you via email.

Online. Register at www.gdconf.com and save an additional \$25.00 when you enter your priority code in the appropriate registration field. Your code may be found above your mailing label. Payment by credit card or check is accepted online.

Please check the boxes below to let us know the types of information that may be of interest

☐ Product and service offerings from outside relevant businesses and organizations.

Mail or Overnight Courier: Mail your completed registration, along with check or credit card payment to: 6DC Registration Department ◆ 600 Harrison Street ◆ 2nd Floor ◆ San Francisco, CA 94107 USA

Fax: Fax your completed two-part registration form, along with credit card payment to 415.947.6020.

Phone: Contact us Monday-Friday, 9am-4pm, PST at 415.947.6135 with credit card payment.

Registering more than 10 attendees? Log onto www.gdconf.com/register/group__discount.html to learn how your group may qualify for a 10% discount.

Cancellations and Substitutions:
If you need to cancel, you may do so for a full refund until February 15, 2002.
Attendees who register prior to, or after the deadline date, who do not cancel in writing by the deadline date are liable and will be charged for the full registration fee. Sorry, no refunds or letters of credit are available after this date. Please far available after this date. Please fav. your cancellation request to 415,947,6020, or mail your request to:
GDC Registration Department
600 Harrison Street, 2nd floor
San Francisco, CA 94,107
Written requests for a downgraded pass

must be received no later than February 15, 2002 for a full refund on the difference of

registration fees between the value of the

original and downgraded pass. Requests

receive a letter of credit for the 2003 GDC issued for the difference in pass values. Upgrade pass requests must be submitted in writing and faxed to 415,947,6020 along

with payment information for the difference

Substitutions are allowed only with the

written permission of the original regis-

trant. Please mail your substitution request to the above address, or fax to

Sorry, no one under the age of 18 is

allowed on the Expo floor.

in value.

415.947.6020.

received after February 15, 2002 will

\$

Membership is \$100

GRAND TOTAL

STEP 4	BUSINESS/LEGAL	47 Post-production
1. Which of the following best describes your company's primary line	39 CEO/President	48 🗌 Platform porting
of business? (select only one)	40 — VP/Executive Manager	49 Localization
GAME INDUSTRY	41 Marketing/PR/Sales	50 Internet service providers
01 Independent 3rd-party game development	42 Analyst/Lawyer/Consultant	51 Web hosting services
02 2nd-party game development with publisher	43 HR/Training Manager/Recruiter	52 Internet/networking infrastructure
03 1st-party game development/publishing	44 🗆 IS/IT Manager	53 Motion capture services
04 Game recruiter/agent	45 - Venture Capitalist/Investor	54 Business (legal, marketing, accounting, PR, etc.)
05 🗆 Online game service provider	PUBLISHING	55 Licensing representation
CONTRACT GAME DEVELOPMENT SERVICES	46 Executive Producer/Producer	56 Professional representation (recruiters, agents)
06 Animation/graphic arts	47 Content Acquisition	57 Packaging production & design
07 ☐ Programming	48 Product Manager	58 Other Services (please specify)
08 Music/sound	49 Other (please specify)	outer services (prease specify)
09 Script writing	is = canon (product opening)	
10 Testing/QA		5. Which products, services or technologies do you
RELATED INDUSTRY	3. What is your involvement in the purchase of tools and services for	recommend, specify, or authorize for support/
	your job? (select all that apply)	optimization? (select all that apply)
11 Animation/graphic arts	01 \square Identify need	01 3D graphics accelerators
12 TV/video/film	02 Evaluate brands or products	02 □ 3D audio
13 Motion capture	03 Recommend brands or products	03 Game peripherals
14 Educational software development	04 Authorize or approve purchase	04 Online game services
15 Tools/Middleware development	05 Not involved	05 ☐ Platform choice
16 Hardware development		06 Other (please specify)
17 🗆 Visual simulations development	4. Which products, services or technologies do you recommend.	ou Conten (prease specify)
18 Multimedia production	specify, or authorize for purchase? (select all that apply)	
19 Corporate software development	HARDWARE	6. On/for which of the following platforms are you developing your
20 Education/research	01 PC/Mac desktops	current/most recent game? (select all that apply)
21 Analyst/venture capital/legal	02 Workstations	ON FOR
22 Other (please specify)	03 Monitors	Windows 95/98/ME 01 □ 08 □
	04 Servers/networking equipment	Windows CE 09
2. Which of the fellowing best describes and the fellowing	05 Consumer audio cards	Windows NT/2000 02
2. Which of the following best describes your job title?	06 Professional Audio/MIDI boards	Linux 03 \(\tag{11} \)
(select only one)	07 Audio recording/editing equipment	Other UNIX 04 12
VISUAL ARTS	08 Consumer graphics accelerators	Mac OS 05
01 Art Director	09 Professional graphics accelerators	Mac OSX 06 14
02 Lead Artist	10 Digitizing/3D scanning equipment	Game Boy 15
03 Lead Animator	11 Video capture/playback boards	Game Boy Color 16
04 Animator	12 Video editing systems	
05 3D Artist/Modeler	13 Data storage/backup	
06 🗆 2D Artist/Texturer	14 DVD/CD-ROM burners	
07 Art Assistant		Cell phone/WAP/Imode 19
PROGRAMMING/ENGINEERING	15 Game input devices (joysticks, etc.)	Other handheld OS 20
08 🗌 Art Technician	16 Productivity input devices (tables, etc.)	CD-ROM 21 \square
09 Director of Development	17 Motion capture equipment	ON FOR
10 Technical Director	18 🗆 Other Hardware (please specify)	DVD 22 □
11 🗆 Lead Programmer	SOFTWARE	Sega Dreamcast 23
12 🗆 Engine Programmer		Nintendo 64 24
13 Al Programmer	19 Commercial game engines	Nintendo Gamecube 25
14 🗌 Tools Programmer	20 Programming environments/compilers	Sony Playstation 26 □
15 Programmer	21 Cross-platform/porting tools	Sony Playstation 2 27 □
16 Network Programmer	22 Programming libraries/middleware	Microsoft Xbox 28
17 🗆 Hardware Engineer	23 3D rendering/modeling/animation	Online game channel (EA.com, etc.) 29
GAME DESIGN	24 \sum 2D draw/paint tools	MMP0G (U0, etc.) 30
18 Creative Director	25 Testing & debugging tools	Web games 31 \square
19 🗆 Lead Designer	26 Programming utilities	Arcade/coin-op 32
20 Game Designer	27 Multimedia/authoring/prototyping tools	Interactive television 33 \square
21 🗆 Level Designer	28 Project/asset management	Other (please specify) 07 🗆 34 🗔
22 🗆 Interface Designer	29 Game security	
23 Writer	30 Image manipulation/photo enhancement	7 W. C. W. L. L. C.
AUDIO	31 Video/film editing	7. What is the budget for your current game project? (select only one)
24 Director, SFX/Music	32 Motion capture software	01 S50 Million or more
25 Composer/Musician	33 Sound composition/editing	02 \(\tag{\$20,000,000} - \$49,999,999 \)
26 Sound Designer	34 Version control software	03 \$10,000,000 - \$19,999,999
27 SFX Engineer	35 Video codecs	04 \$5,000,000 - \$9,999,999
PRODUCTION	36 Audio codecs	05 🗆 \$3,000,000 - \$4,999,999
28 Executive Producer	37 Music libraries	06 \$2,000,000 - \$2,999,999
29 Producer	38 Stock footage/clip media	07 \$1,000,000 - \$1,999,999
30 Associate Producer	39 3D model libraries	08 \$500,000 - \$999,999
31 Project Lead/Manager	40 ☐ SFX libraries	09 \$100,000 - \$499,999
32 🗆 Video Director	41 🗌 Other Software (please specify)	10 Less than \$100,000
33 Asset Manager		CTEDE
34 🗆 Editor	SERVICES	STEP 5 Game Developer magazine*
35 Localization	42 Contract art/animation	YES, I wish to receive/continue to receive a free subscription to
36 Licensing	43 Contract programming	Game Developer.
37 QA/Tester	44 Contract music/sound	□ No, I do not wish to receive/continue to receive Game Developer.
38 Documentation Development	45 Contract testing & QA	
	46 Disc replication	Signature Date

Signature

Date

Game Developers Conference

600 Harrison Street • San Francisco, CA 94107 • USA

Conference: March 19-23, 2002 Expo: March 21-23, 2002

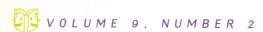
San Jose, CA

Register by February 5, 2002

and **save up to 30%** on GDC 2002 Conference Passes.

Register online using the priority code above your address and **save an additional \$25**.





CONTENTS

FEBRUARY 2002

FEATURES

30 Play by Play: Effective Memory Management

Five programmers from Electronic Arts-Tiburon show the in and outs of building an all-Madden-class memory manager.

> Brian Hixon, Daniel Martin, Rob Moore, Greg Schaefer, and Richard Wifall

36 Four Ways to Use Symbols to Add Emotional Depth to Games

The established practice of screenwriting holds some clever lessons for today's ambitious game designer. Learn how verbal and visual symbols can heighten the player's emotional involvement in gameplay.

David Freeman

54 Postmortem: Mythic Entertainment's DARK AGE OF CAMELOT

Matt Firor

Discover how Mythic's solid game development experience, an existing code base of proven technology, and an unwillingness to limit the bounds of the imagination came together in this successful MMORPG.

COLUMNS

- 18 THE INNER PRODUCT Jonathan Blow Mathematical Growing Pains
- 25 ARTIST'S VIEW

 Making Trees Work

 Hayden Duvall
- 56 SOAPBOX Hal Barwood
 The Envelope, Please

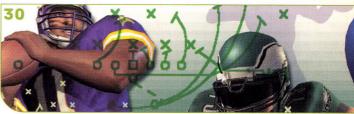
COVER

This month's cover was inspired by game art from DARK AGES OF CAMELOT. It was created by Mythic Entertainment art guys C.J. Grebb, Mat Weathers, and Lance Robertson using 3DS Max and Adobe Photoshop.









DEPARTMENTS

- 2 GAME PLAN Jennifer Olsen
 - The Joys of Self-Regulation **SAYS YOU**
- Letters from our readers
- 6 INDUSTRY WATCH Daniel Huebner
 - Christmas and end-of-year news and blues

 PRODUCT REVIEWS
- Lightwave, Spectra Foo, Accurev, and Rhythm 'n' Chords
- 16 PROFILES Warren Spector
 Warren Spector's Sid Meier!
- 51 CREATIVE CAREERS
- 53 GET EDUCATED

55 ADVERTISER INDEX

Game Developer (ISSN 1073-922X) is published monthly by CMP Media LLC, 600 Harrison St., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian GST: 124513185. Subscription Rates: Subscription rate for the U.S. is \$49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: \$69.95; all other countries: \$99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. Postmaster: Send address changes to Game Developer, P.O. Box 1274, Skokie, IL 60076-8274. Customes Service: For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5912. Send payments to Game Developer, P.O. Box 1274, Skokie, IL 60076-8274. For back issues write to Game Developer, 1601 W. 23rd St. Ste. 200, Lawrence, KS 66046-2703. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 841-2624. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate Game Developer on any correspondence.

www.gdmag.com



GAME PLAN

DLETTER FROM THE EDITOR

The Joys of Self-Regulation



hat kind of kid were you? Did your mom have to nag you endlessly to clean your

room, your constant refusals erupting into all-out wars? Or perhaps you were cheerful and obedient, always cleaning your room as you were told. Sometimes, if you did it without your mom having to ask, she might give you a cookie.

The game industry got its cookie in December when the Federal Trade Commission issued its latest report to Congress on "Marketing Violent Entertainment to Children," a follow-up to a report submitted in September 2000 which criticized marketing practices in the music, film, and videogame industries. The conclusion was that since the first six-month follow-up report released in April 2001, the film and videogame industries have continued to improve their rooms' tidiness with respect to diverting their marketing of violentthemed or otherwise objectionable entertainment away from minors. (The music industry, apparently, is still running away from Mom, screaming and slamming doors, and seemingly oblivious of the fact that Mom usually wins in the end.)

The self-regulatory efforts of the game industry's trade organizations, in particular the Interactive Digital Software Association and the Entertainment Software Rating Board along with its Advertising Review Council, are commendable for willfully assuming responsibility for industry marketing practices while rightfully continuing to defend our First Amendment rights. What they've so deftly realized is that when you address one critique purposefully and with measurable success, you fend off outside agencies who would regulate this industry without its best business and creative interests in mind. Successful self-regulation also helps to dilute criticisms lobbed at other aspects of our trade. But we're not out of the publicopinion woods yet.

The most important thing for our industry to do now is to continue to defend

itself against errors of fact in public opinion, such as the persistent and pernicious misperception that videogames are predominantly made for and played by children. When Australia's Office of Film and Literature Classification devised its first videogame rating system in 1994, "...concerns were expressed about the interactive nature of computer games and the possible adverse effects on children, who were seen as the primary target audience for computer games." (OFLC Discussion Paper, "A Review of the Classification Guidelines for Films and Computer Games," 2001). Later guidelines established a 15-and-over rating for some games, but decreed - in a triumph of ambiguity — "the stronger computer games are banned." (OFLC "Guidelines for the Classification of Computer Games," 1999).

The same week the FTC released its updated report in the U.S., copies of clearly mature-themed games such as GRAND THEFT AUTO 3 and GHOST RECON were reportedly being whisked from store shelves in some Australian jurisdictions, at the height of the Christmas shopping season, in order to undergo reclassification. The head of Sony Computer Entertainment Australia can tell a reporter for the Sydney Herald Sun that half of Playstation 2 owners are over 30 years old, yet the country's regulatory body concluded just a few years earlier that there was no need for a mature rating for games as there was for film, because games were for children. Clearly there is a lot of confusion coming out of and real money evaporating into this non-self-regulatory system.

Lessons abound about the virtues of successful self-regulation and the perils of yielding control to outside interests. The U.S.-based industry has fortunately been rewarded with praise for its efforts so far, but now is not the time to rest on our gossamer-thin laurels, nor will it ever be.

Sennfer

GameDeveloper

Harrison Street San Francisco CA 9/107 1, /15 9/7 6000 6, 415 9/7 609

Publish

Jennifer Pahlka jpahlka@cmp.com

EDITORIAL

Editor-In-Chief

Jennifer Olsen jolsen@cmp.com

Managing Editor

Tor Berg therg@cmp.com

Production Editor

Olga Zundel ozundel@cmp.com

Art Director

Audrey Welch awelch@cmp.com

Editor-At-Large

Chris Hecker checker@d6.com

Contributing Editors

Daniel Huebner dan@gamasutra.com
Jonathan Blow jon@bolt-action.com

Hayden Duvall hayden@confounding-factor.com

Advisory Board

Hal Barwood LucasArts Ellen Guon Beeman Beemania Andy Gavin Naughty Dog Joby Otero Luxoflux Dave Pottinger Ensemble Studios George Sanger Big Fat Inc. Harvey Smith Ion Storm

Paul Steed WildTangent ADVERTISING SALES

Director of Sales & Marketing

Greg Kerwin e: gkerwin@cmp.com t: 415.947.6218

National Sales Manager

Jennifer Orvik e: jorvik@cmp.com t: 415.947.6217

Senior Account Manager, Eastern Region & Europe

Afton Thatcher e: athatcher@cmp.com t: 415.947.6224

Account Manager, Northern California & Southeast

Susan Kirby e: skirby@cmp.com t: 415.947.6226

Account Manager, Recruitment

Raelene Maiben e: rmaiben@cmp.com t: 415.947.6225

Account Manager, Western Region & Asia

Craig Perreault e: cperreault@cmp.com t: 415.947.6223

ales Associate

Aaron Murawski e: amurawski@cmp.com t: 415.947.6227

ADVERTISING PRODUCTION

Vice President, Manufacturing Bill Amstutz

Advertising Production Coordinator Kevin Chanel

Reprints Stella Valdez t: 916.983.6971

Reprints Stella Valdez t: 916.983.69/1

GAMA NETWORK MARKETING

Senior MarCom Manager Jennifer McLean

Marketing Coordinator Scott Lyon

Audience Development Coordinator Jessica Shultz

CIRCULATION



Group Circulation Director Catherine Flynn
Circulation Manager Ron Escobar
Circulation Assistant Ian Hay
Newsstand Analyst Pam Santoro

Game Developer is BPA approved.

SUBSCRIPTION SERVICES

For information, order questions, and address changes

t: 800.250.2429 or 847.647.5928 f: 847.647.5972 c: gamedeveloper@halldata.com

INTERNATIONAL LICENSING INFORMATION

Mario Salinas

t: 650.513.4234 f: 650.513.4482 e: msalinas@cmp.com

CMP MEDIA MANAGEMENT

President & CEO Gary Marshall

Executive Vice President & CFO John Day

President, Technology Solutions Group Robert Faletra

President, Business Technology Group Adam K. Marder

President, Healthcare Group Vicki Masseria

President, Specialized Technologies Group Regina Starr Ridley

President, Electronics Group Steve Weitzner

Senior Vice President, Business Development Vittoria Borazio Senior Vice President, Global Sales & Marketing Bill Howard

Senior Vice President, Human Resources &

Communications Leah Landro

Vice President & General Counsel Sandra Grayson
Vice President, Creative Technologies Philip Chapnick



GamaNetwork

Register now at http://www.gdconf.com for Microsoft® Developer Day at the Game Developers Conference Wednesday, March 20, 2002

Learn about cutting-edge technologies like

- DirectX_® High Level Shading
- DirectX Graphics Performance Optimization
- Material Surface Physics for Shader Writers
- · Getting the most from D3DX
- The new 3-D Graphics Accelerator Generation of 2002
- Shader Optimization with DirectX 8

Microsoft DIRECT

3D SOUND PLAY MUSIC SHOW INPUT

Also be sure to attend these info-packed Sponsored Sessions on **Thursday, March 21!**

- Developing Games for PocketPC and the SmartPhone platform
- Managed DirectX Learn about the new managed layer in DirectX
- Optimizing with Microsoft
 Visual C++
 7.0 Find out what it takes to get the best performance from your compiler
- Gaming and the Web How Emerging Technologies may enable Game Developers
- Getting the most from D3DX Get an in depth tour of the D3DX library
- DirectPlay Networking Layer in Depth Get an in depth view of DirectPlay's architecture from its developers

Get the latest on DirectX at http://msdn.microsoft.com/directx/

© 2001 Microsoft Corporation. All rights reserved.
Microsoft, DirectX, and Visual C++ are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
01101

Microsoft DIRECTX®

SAYS YOU

A FORUM FOR YOUR POINT OF VIEW. GIVE US YOUR FEEDBACK...



Game Engines, or Are They?

object to Andrew Kirmse and Daniel Sanchez-Crespo's classification of NetImmerse and Alchemy as "game engines" in "Test Drive: On the Open Road with Two of Today's Most Powerful Game Engines" (December, 2001). They don't qualify as engines at all; they're component sets.

> Anonymous via e-mail

DANIEL SANCHEZ-CRESPO RESPONDS: The term "game engine" is pretty slippery. Traditionally, it has referred to "closed solutions," which allowed the developer to concentrate on content creation. In this respect, NetImmerse and Alchemy should be better called "game development toolkits." Both Andrew and I took care that this idea was clearly stated in our reviews.

Still, this semantic precision is a double-edged sword. Being "traditional" engines, Quake & Unreal should allow teams to concentrate on content, right? Still, when Valve used the Ouake 2 engine to create HALF-LIFE, some components were reworked or written from scratch. Does that make Quake 2 less of an engine? Now, consider HIRED GUNS, a game built on top of Unreal. What's so interesting about it? Well, it's a real-time strategy game, clearly not what the people at Epic had in mind when they coded their software. The whole interface was replaced, and I can guess lots of AI/logic code needed some major reworking. Should we change the naming of those packages in that case?

Most teams working on classical game engines are in fact using a toolkit approach: analyzing the available components and reworking those that need it. Should we totally drop the "engine" term if even the most representative products violate the definition? Lots of precision can make it impossible to classify items into groups, as each item has specific features that make it unique. Being practical (and, yes, adding some

imprecision), we can consider Ouake and NetImmerse members of a same family which, for historical reasons, we can call engines. You will always have closed engines, which will offer a faster time to market, and toolkits, which give us greater flexibility. In the end, all these products are nothing but close relatives, so the incurred imprecision in the naming is, in my opinion, justified.

Kudos for "The Inner Product"

have just read Ion Blow's first "Inner Product" column ("Mipmapping, Part 1," December 2001). I am currently on a game project and the information he has supplied is more than enough to get my team thinking on how we could implement a better mipmapping algorithm.

> Steve Marth via e-mail

Don't Forget to Gamma Correct

'm so glad to see that Jonathan Blow's Inner Product" column has picked up the technical torch at Game Developer ("Mipmapping, Part 1," December 2001).

Besides the ringing mentioned in the article, another effect usually ignored by game programmers (and almost everyone else) is the effect of gamma correction (or lack thereof) on PC monitors. Since I didn't see this effect mentioned in the article, here's a brief rundown. Imagine your base texture is a checkerboard, 2×2, two black and two white squares. The pyramid derived from this texture is a 1×1 gray, usually stored as 0.5. However, the answer should really be more like 0.73, if you factor in gamma correction for CRT monitors (LCD monitors screw up the equation usually, as their response is different).

Who cares? Well, in this case, if you use 0.5 then the object is far away and appears dim, and as it comes closer, it gets brighter. Not a huge deal, but it's so easy

to avoid if you're precomputing mipmaps using elaborate filters (such as the article describes). You might as well get gamma correct, too.

If you're smart, you do all this with at least 12 bits of precision per channel, to avoid banding (Jim Blinn talks about this precision problem with gamma conversion in his books).

> Eric Haines via e-mail

JONATHAN BLOW RESPONDS: Actually, in an upcoming column, I talk about gamma correction. Rumor has it that the next chip design from A Major 3D Accelerator Maker has deep enough channels and a versatile enough RAMDAC or page copier that you can just keep the frame buffer in light-linear space and exponentiate everything after the whole frame is drawn. This has good connotations for lighting (in other words, it becomes basically free to actually do lighting at the proper falloff rate).

Teaching Games

enjoyed Celia Pearce's "Learning Curves: The Present and Future of Game Studies" (Soapbox, December 2001).

Six years ago I proposed a videogame programming course to Paloma College, a community college in San Marcos, Calif., which I have been teaching parttime for the past five years. We recently decided to expand our program by offering two new courses. The first will be offered this spring semester: "An Overview of the Videogame Industry." The second class to be started next year will be a game programming course. We will be offering a videogame specialist certificate, but hope to expand it to an A.A. degree in videogame programming.

> Ed Magnin via e-mail



Let us know what you think: send e-mail to editors@gdmag.com, or write to Game Developer, 600 Harrison St., San Francisco, CA 94107



INDUSTRY WATCH

THE BUZZ ABOUT THE GAME BIZ | daniel huebner

Christmas console launches. The month leading up to the all-important

holiday period saw the start of the second round the ongoing console wars, as Xbox and Gamecube finally made their long-awaited debuts just days apart. While both new consoles initially seemed to sell out as quickly as stores could stock them, in the weeks

following the launches it was difficult to decipher the numbers to conclude who actually had the better launch.

Nintendo claimed that sales of nearly 600,000 consoles in the first 15 days of Gamecube availability made the launch the most successful console debut ever. While Microsoft had just half as many units ready for launch and hadn't publicized its sales numbers at press time, the company claimed the most successful launch title for Xbox, asserting that Bungie's HALO was out-selling Nintendo's LUIGI'S MANSION. Both companies hoped to have more than 1 million consoles in consumers' hands by the end of the year.

The other consoles. Sega was moving the final Dreamcasts out the door after cutting the price on remaining consoles

from \$79.95 to \$49.95. New pricing has moved the discontinued machine at a brisk pace, pushing Dreamcast sales past the 10 million mark. Said Sega's Peter Moore of the toolate sales surge, "Ironically, we now wish we had more."

Sega's post-Dreamcast recovery is moving along slowly, as the company posted a \$169 million loss in the six months ended in September. Sales were down 18 percent from the previous year.

Sony set a new price for Playstation 2 at the end of November, but the new rate

didn't extend to North America, at least for the duration of 2001. The 15 percent price reduction was credited to reduced production costs rather than as a response to console launches from competitors Nintendo and Microsoft.



Microsoft proclaimed HALO the best-selling console launch software for the holiday shopping season.

SUPER MONKEY BALL proved to be a pop-

ular Gamecube launch title.

Bleem emulator

gives up. Bleem, the company that made Playstation emulators for PC, Macintosh, and Dreamcast, has shut its doors after protracted legal battles with Sony over copyright infringement. Sony first sued Bleem over its products in May 2000. Bleem counter-sued, claiming Sony was exercising an illegal monopoly over the videogame industry.

Nvidia replaces Enron on S&P 500.

The sudden demise of energy firm Enron Corp. was good news for chip maker Nvidia. Standard & Poors announced at the end of November that Nvidia would replace Enron in the prestigious Standard & Poors 500 Composite Index.

Square CEO quits after poor showing by Final Fantasy. Game software

maker Square announced that president and chief executive officer Hisashi Suzuki would resign after the company reported its worst-ever loss for the first half of its fiscal year due to a disappointing showing by *Final Fantasy: The Spirits Within.* Square reported a group net loss of \$106.8 million for the

six months ended September 30. Chief operating officer Yoichi Wada was scheduled to take the top position on December 1, while Suzuki remains as the chairman. The film generated box-office

revenue of about \$30 million in the U.S. market, well below the targeted \$80 to \$90 million, and interest among Japanese consumers has also been weak.

Konami group net profit plunges.

Konami announced a drop of 78.3 percent in group net profit in the first half of the company's fiscal year. Most of of the deficit was attributed to profit shortfalls for the company's Yu-Gi-Oh card game. Group net profit dropped to \$20.74 million from \$94.64 million for the six months through September, despite total sales rising nearly 20 percent to \$725.48 million. Growth in sales of its videogames business, however, could not offset a sharp decline in operating profit in its card game business.

Interplay reports few bright spots in third-quarter financials. Interplay reported net revenues of just \$4.2 million for its fiscal third quarter, a drop of 87 percent from last year. The net loss for the period was \$20.6 million, a disappointing result after reporting net income of \$0.1 million in the same period last year. Most of the drop can be attributed to failing to ship games; Interplay didn't have any new titles in the third quarter and had shipped just seven for 2001 as of mid-December.

Interplay released a total of 26 titles in 2000.

CALENDAR

D.I.C.E. SUMMIT

HARD ROCK HOTEL Las Vegas, Nev. February 28–March 1, 2002 Cost: variable www.interactive.org

GAME DEVELOPERS CONFER

ENCE 2002
SAN JOSE CONVENTION CENTER

www.gdconf.com

San Jose, Calif. March 19–23, 2002

Cost: \$195-\$1,950 (early-bird discounts available)

6



lust list





RenderWare, the secret behind the chart toppers...

...and this is just the tip of the iceberg!



© 2002 Criterion Software Ltd. All rights reserved. Criterion and RenderWare are registered trademarks of Canon Inc. All other trademarks methioned herein are the property of their respective companies.



Austin • Derby • Guildford Paris • Tokyo

GameDevelopers

Conference

March 19-23, 2002 • San Jose, CA

Make better games

Learn directly applicable techniques in the core disciplines of game development.

Tracks:

Visual Arts

Game Design

Programming

Audio

Business & Legal

Level Design

Production





PRODUCT REVIEWS

Newtek's Lightwave 3D 7b

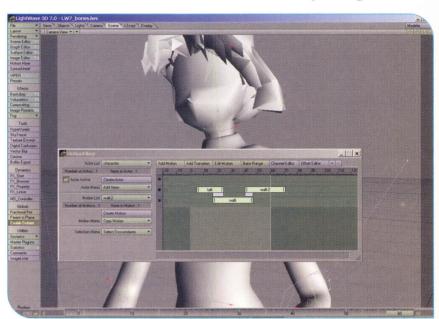
by sergio rosas

ewtek's latest release of its 3D modeling and animation package is crammed full of goodness. So what do you get for the upgrade price? You get some new features, some great enhancements to the established ones, and an occasional simple but elegant refinement.

Motion Mixer. Finally, Lightwave 3D gets its very own nonlinear animation system. Lightwave users have been waiting for a long time, and it's finally here: Motion Mixer. With this tool, you can define a character and then build a library of its motions. Once you've identified the motions (such as a run cycle and a walk cycle) you can edit them together and easily blend from one to another. Additionally, Motion Mixer allows you to set pre- and post-behaviors for motions (such as repeat and oscillate), scale, load, and save entire hierarchies of motions. It also has a nice motion mapper tool that lets you load motions from a different character.

Motion Mixer is a great tool, but after I used it for a little while, it left me wanting more. Unfortunately, Motion Mixer restricts its motions to a minimum of five frames. This means that you can't work with simple poses. Motion Mixer also lacks the ability to load in standard motion capture files. Nevertheless, if you still do character animation in Lightwave, this feature alone is worth the price of the upgrade.

Spreadsheet editor. Another great new feature in Lightwave is the spreadsheet editor. This view looks similar to the scene editor, except that you can expand it spreadsheet-style with tons of item properties. All of these properties can be selected en masse and tweaked. You can reset them all to a new value or add/subtract/multiply an offset to every one.



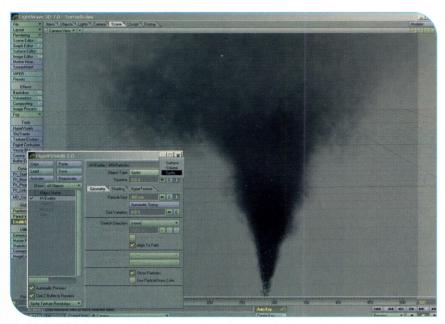
LIGHTWAVE 7B. Motion Mixer is Lightwave's long-awaited new nonlinear animation system.

You can make a massive change, preview it in real time, and if you like it, you can apply it to the scene. It also has a timeline, just like the scene editor, where you can move and size motions. You can make custom workspaces and save them with your scene. You can sort by any property and use its powerful filter for the items list. It took me all of two seconds to figure out and love this feature. I really wanted this feature when I had a scene with 78 lights that needed lens flares. Just for fun, I loaded that old scene up and was sorely disappointed to find that the speadsheet editor doesn't do lens flare properties. In short, the spreadsheet editor adds some extremely useful functionality, but you still might have to resort to a text editor every once in a while.

Sprites. Some might not consider this feature to be a big deal, but coming from game land, sprite-based particles are a huge deal to me. Both hypervoxels and volumetric lights have sprite options now. There's nothing to it, just click the sprite checkbox and let Lightwave take over. It renders the sprite version of your hypervoxel internally and assigns it to each particle. (Of course, you can also assign your own texture if you want.) This is a very useful option for game developers. You can see the particle sprites animating in layout in real time. Even though it's not exactly the sprite that being rendered, it's great for tweaking timing. When rendered, both the voxel and volumetric light sprites look just fine, even when flying through them. And they render a lot faster than in pre-

SERGIO ROSAS | Sergio is currently acting as lead artist for Thief 3 at Ion Storm. He can be reached at srosas@ionstorm.com.

www.gdmag.com



LIGHTWAVE 7B. Sprite options for hypervoxels and volumetric lights make these viable effects for game animators.

vious versions. This makes hypervoxels actually usable for me.

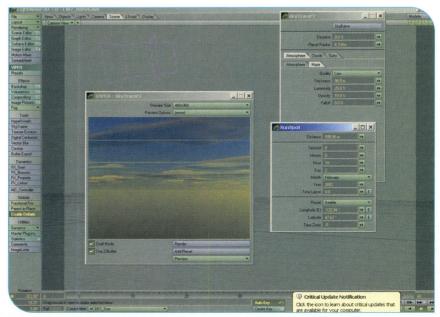
Rendering speed. Speaking of speed increases, Newtek added quite a few with Lightwave 7b. Besides being able to use sprites to speed up hypervoxels, you can also bake a hypervoxel cloud to get great rendering speed increases. Radiosity got a new "backdrop only" setting that renders faster even if it doesn't look quite as good. Also, a new "shading noise reduction" global illumination option can make the low-end area light and radiosity settings look less grainy while rendering much faster than the higher settings.

Enhanced particle system. Lightwave comes with an integrated particle system, and (although it might not be as good as the third-party systems) it's getting much better. With this release, the particle engine got interparticle collisions, particle respawning, and collision spawning. This means that you can simulate rain — each particle spawns little splash particles when it hits the floor. You're now able to attach an object or hierarchy to a particle, so you can make giant meteors crashing into each other and breaking into tiny pieces. Particle

emitters also got a bunch of new nozzle types, such as sphere, cone, and object vertices. Newtek also added a bunch of new wind types, such as turbulence, vortex, and explosion. The wind types are powerful and have a great visual representation that makes them a snap to use. Everything about the internal particle system feels a lot more integrated than in previous versions — they even interact with motion designer soft bodies.

Graph editor improvements. With every new version of Lightwave, the layout graph/curve editor gets better and better. This time Newtek added an OpenGL interface, which the company claims is able to do hundreds of curves at once. It does feel much smoother and faster than it ever did before. Another new feature that's a great time-saver, the graph editor will now open with the curve for the current item selected. Unfortunately it doesn't open all of the curves if you have multiple items selected. Some other enhancements include match footprint, key reduction, curve filtering, and key bins.

Modeler enhancements. Compared to the Layout interface, the Modeler got very few new features. Newtek added a rounded box primitive and a more useful curve tool. The new face collapse command lets you select adjacent polygons and collapse them all to a single point. This is a real time-saver for optimizing models. You can also now airbrush between morphs. (Neat, but it's no Artisan.) Additionally,



LIGHTWAVE 7B includes Sky Tracer, a powerful and elegant feature for creating atmospheric effects.



being able to save a single layer as an object and to flatten all layers will come in handy to game animators.

LScript Commander. This new tool keeps a list of all the behind-the-scenes commands that are executed when you're clicking around the Layout interface. You can create macros easily by copying and pasting selected commands into the work area and hitting the Execute button. You can also install your own macro button from there. LScript Commander is not as slick its Maya or 3DS Max counterparts, but it's a start.

LIGHTWAVE 7b

STATS

NEWTEK

5131 Beckwith Blvd. San Antonio, TX 78249 (210) 370-8000

www.newtek.com

PRICE

\$2,495 (upgrade price is \$495) SYSTEM REQUIREMENTS

Windows: Windows 98/ME/2000 (Service Pack 2)/NT 4 (Service Pack 6a), TCP/IP network protocol installed, 128MB RAM. Macintosh: PowerPC Processor (G3 or higher recommended), Mac OS 9 or Mac OS X (recommended), 384MB RAM for Mac OS 9, 128MB RAM for Mac OS X. All systems require 32MB available hard drive space, CD-ROM for installation, and a minimum screen resolution of 800×600 pixels.

PROS

- 1. Motion Mixer nonlinear animation tool
- 2. New spreadsheet editor
- 3. Sprite-based hypervoxels and volumetric lights

CONS

- New features (Motion Mixer, spread sheet editor, vertex paint) seem disconnected and inconsistent with the rest of the package
- 2. Motion Mixer nonlinear animation tool can't use single-frame poses
- 3. No motion capture import

Sky Tracer 2.0. Sky Tracer got a facelift for Lightwave 7b. It's now fully integrated into Viper and seems easier than ever to use. It has a great Suns feature, which lets you type in the month, day, year, geographical location, and time, then it instantly pops out a great looking sky. It also has a Baker feature that easily exports a skybox — another old plug-in made usable for game developers.

Toon shading. For those of us who like stylized renders, Lightwave 7b ships with BESM (Big Eyes Small Mouth), a great shader for cartoon rendering. It has plenty of great options such as unlimited zones of shading and variable blending between zones. Each zone has opacity, brightness, and saturation settings. Specularity options, gradient overlays, edge options, and much more make this my favorite toon shader.

To Buy or Not to Buy?

ightwave 7b has some very useful and long-awaited new animation features that might seem a little rough around the edges to some. Those accustomed to other character animation packages will probably not be tempted to make the switch. Still, these new features, along with all the great refinements made to existing features, are a must-have for veteran Lightwave animators. In the realm of special effects, Lightwave's hypervoxels are top notch and continue to refine. The addition of sprites makes Lightwave game developer friendly. For the low-polygon modelers, this version of Modeler has few new features, but it continues to be the best polygon modeler for the money. If you are doing UV mapping in Lightwave, you should upgrade to 7b. For anyone doing cinematics, or for rogue contract artists, I would make this my tool of choice. To cut a long story short, the "b" in Lightwave 7b stands for "bang for buck."

ACCUREV 3.0.1

by michael saladino

R evision control software is the safety net that allows multiple programmers to work together on the same

code base without trampling on each other. Most systems fall into one of two camps: either a low-cost solution, such as Microsoft's Visual SourceSafe, or a massive system requiring a team of experts to keep it running, such as Rational's ClearCase. Accurev is trying to carve a niche for itself between these two extremes.

I was able to get my trial copy up and running within minutes of downloading it. The server setup was simple and painless. I had slightly more difficulty when working with the client side, but that's undoubtedly because the client is where the greatest amount of functionality is located. My test case was to take my current game project and copy it to Accurev. More than 2,000 files were uploaded into the system in a matter of minutes, my first indication that Accurev is extremely fast. Soon after this, I was able to get complete source for my client machine with all the basics, including difference, history, and checkout control. My next task was to time basic operations, and what I found was impressive. From checkouts to history differences, Accurev performed standard functions many times faster than the Visual SourceSafe equivalents.

However, where Accurev really shines is in its solutions to the subtle "gotchas" that often slow production when one uses low-budget software. One of the classics is when a programmer checks in a large group of files at the same time that another programmer is getting the latest code. The person getting code will only receive part of the total check-in, which in most cases will render the build unable to be compiled. Accurev disallows this by keeping transactions such as large check-ins from applying publicly until it is complete.

A more severe test that I ran was to begin a check-in of multiple files and then disrupt the transaction by pulling the plug on the Accurev server (a little extreme, but I'm testing reliability here). I know from experience that losing power during a Visual SourceSafe transaction can be a dangerous event, resulting in corruption of the database. Accurev, however, was able to survive nicely. When I restarted, everything was

www.gdmag.com

functioning and the check-in process could be restarted on the client with no loss of data.

My only serious concern with Accurev is its interface design. While certainly easier to use than industry standards such as ClearCase, it still lacked the simplicity of Visual SourceSafe. The supposedly simple matter of bringing new files into the database was something that kept me stumped for far too long. Not surprisingly, I found it easier to integrate SourceSafe with Microsoft's Visual Studio. And while a cursory inspection of the software might remind you of SourceSafe's look and feel, its internals are definitely different and do require time to learn. I found that by the end of a couple hours of testing, I was beginning to feel more comfortable.

Accurev 3.0.1 is available for Windows 95/98/M/NT 4.0/2000/XP and an impressive host of other, more obscure platforms. The evaluation version is not time-limited, but is limited to two users. Accurev will provide interested parties with a price quote for extended licenses.

For my next project, I will certainly experiment more with Accurev to determine whether it would be a worthwhile change. From my initial tests, it's a promising new option for software developers no matter what the size of the project.

* * * | Accurev 3.0.1 | Accurev www.accurev.com

Michael Saladino is senior programmer at Presto Studios in San Diego.

METRIC HALO'S SPECTRA FOO

by gene porfido

etric Halo appeared on the audio radar a few years ago with a dazzling new program called Spectra Foo. "Spectra who?" you ask. That's Spectra Foo and it's an RTAS and MAS plug-in with an incredible array of functions that make it as easy to remember as its name.

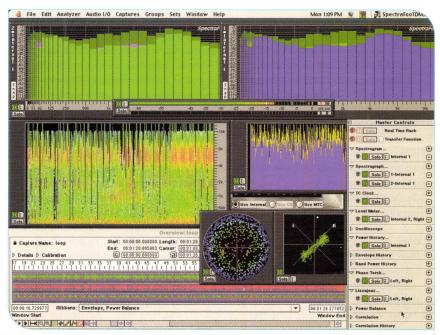
Since its introduction, Spectra Foo has become the Macintosh standard for every imaginable audio-monitoring or test-job function one could dream of, and it has matured well in later versions. Some call it the Swiss Army knife of audio applications. And for professionals from mastering engineers to sound designers, it's like having a toolbox full of ultra-accurate meters, an expensive oscilloscope, and waveform displays built into your Mac.

There are two versions of "Foo," as it is commonly called, including the \$400 standard version and the \$800 Spectra Foo Complete. Each version is capable of high-resolution metering and measuring spectral analysis, phase correlation, waveforms, power, envelope, and spectral histories, as well as a number of highly configurable input/output configurations for matching or comparing audio signals.

While the standard version of Foo is extensive on its own, the complete version adds quite a few top-end features that augment its already impressive functions. There's a world-class signal generator that performs Pink and White noise, Burst, FFT, and direct-to-audio signal generation at 24 bits and with up to nine simultaneous sine wave sweeps. A Transfer function measurement system can handle equipment and test verification, acoustic correction, and time-alignment for studios or labs, and takes direct measurements,

including frequency and phase, of acoustic and equipment signals. Add the Capture and Analysis system to record and open sound files directly into Foo, and you can begin to see how incredibly intricate this software package is.

The program runs well on a PowerPC 604e chip at 180MHz, but I'd recommend a G3 or G4 for heavy-duty work. It's very easy to set up as a plug-in inside of Pro Tools or Digital Performer (RTAS and MAS audio systems, respectively), but can also be opened as a stand-alone application. Setting up Foo to take audio inputs from your Mac I/O, CD, or other input source is no harder than it would be with your favorite DAW. With sufficient processing power, multiple instances can be opened simultaneously to monitor a mix or any number of individual tracks within your DAW program. Window Sets, which are customizable and can be assigned to a hotkey for quick access, offer a snapshot feature of every parameter for instant recall or retrieval as a preset. And a new Link feature keeps all of your parameters from any set of instruments and meters associated with as many groups of instruments as needed.



SPECTRA FOO gives sound engineers a stable and feature-laden toolbox for analyzing digital audio.

Buy an engine. Gain a partner.



When you license NDL's NetImmerse 3D game engine and development toolkit, you get much more than great code. You gain a development partner with engineers dedicated to doing everything possible to ensure the success of your title.

NetImmerse is the only engine that can deliver all of this:

- A complete toolset that accelerates your art pipeline.
- Optimized performance across all platforms.
- Full source code for all libraries and tools.
- Responsive support from those who develop the engine.

Great software. Solid partner. Proven success.

NetImmerse

The fastest path from conception to completion

925.556.9995 • sales@ndl.com



The Elder Scrolls III: Morrowind courtesy of Bethesda Softworks.
Star Trek[®] Bridge Commander courtesy of Totally Games.
Dark Age of Camelot courtesy of Mythic Games.
Call of Cthulhu: Dark Corners of the Earth courtesy of Headfirst Productions.
Pinocchio 3001 courtesy of CineGroupe Interactive.

Spectra Foo is a name worth remembering if audio has any importance in your life. Whether your mastering the next Pink Floyd record or just making sure your game's sound effects are panned and in phase, this is an application that will come in handy in more ways than you'll ever imagine. Count on superlative support from Metric Halo's Dan Metivier, along with the company's incredible (and growing) lineup of exceptional audio software and hardware, and you'll soon be singing the praise yourself. This is what great software is all about.

* * * * * | Spectra Foo | Metric Halo www.mhlabs.com

Gene Porfido has been making music for over 25 years and game sound design for nearly a decade. He is currently an independent sound designer and composer in San Francisco.

MUSICLAB'S RHYTHM 'N' CHORDS 2

by todd m. fav

hythm 'n' Chords is a plug-in for Cakewalk and Steinberg's Cubase series sequencers that takes advantage of

A/C^s AIC AÆ A A , A S25 Back G.D , B'AD A/C^s A/C^s AÆ IA BID BIC AÆ , A B'ID BIE G.D P12 Sask A/C X A/C³ AÆ "A " Gm/D G_n/F G_m/E "BID BIC "BID P12 Basic R A BIK

RHYTHM 'N' CHORDS offers an interface that will be familiar to musicians.

proprietary performance-modeling technology. Performance modeling focuses on emulating the nuances of human instrumentalists through computer technology. Rhythm 'n' Chords models rhythm guitar performances specifically. Creating realistic guitar parts by hand via MIDI can be an extremely time consuming activity. Rhythm 'n' Chords alleviates MIDI musicians from having to stress over those allimportant guitar parts.

Musiclab delivers the plug-in directly from their web site, and here are four different versions for download. Rhythm 'n' Chords Lite is a free download available as a demo. Rhythm 'n' Chords 2 Standard, Rhythm 'n' Chords Pro, and Rhythm 'n' Chords Pro Gold all support an increasing number of features. Musiclab also offers libraries of strumming and picking patterns programmed by a team of engineers and musicians for use with the Rhythm 'n' Chords plug-in, but most of the libraries are included with the Gold version of Rhythm 'n' Chords. There are currently 18 different libraries featuring popular styles such as jazz, oldies, blues, rock, world, and funk. These libraries provide musicians with

> starting points for creating rhythm guitar parts in a particular style.

For those of you seeking the ultimate in guitar performances, there is still only one tried and tested method for capturing truly inspiring work — that is, of course, recording a guitarist. For the musician working on a budget or composer working on a preproduction version or a recording, however, this plug-in can be a real timesaver. After using Rhythm 'n' Chords to lav the groundwork for the rhythm guitar parts, the arranger is free to further "humanize" the content. The plug-in makes it possible to create hours' worth of work in mere minutes.

With more and more games taking advantage of interactive MIDI-based music production, a tool that can aid in the production of realistic musical performances via MIDI is a welcome addition to any game audio professional collection. This plug-in can help raise the production value of a piece while still allowing for the advantageous small file size associated with MIDI.

Knowledge of the guitar helps, but is not required. The interface is similar to a jazz chart. You select chords from a list of supplied voicings and enter them into the measure. Then you either select a rhythm pattern from a library or use one that's been custom created. The arpeggiation time of the strum, chord voicing, and tone can all be adjusted quickly via an easy-to-use interface. Chord voicings, for example, are spelled out on a miniature graphical keyboard, so even those with no exposure to guitar playing can create the right harmonies. You can even control the amount of muting without having to switch to a different guitar patch. This is done using Rhythm 'n' Chords' virtual sliders. There is also a manual play feature that allows the plugin to be used with a keyboard. This way, chords can be voiced by playing them on the keys while the plug-in plays through the user-selected rhythm patterns. This feature is especially useful for those who like to spend time with their hands on the keyboard and off the mouse.

Note that Rhythm 'n' Chords is only available for use in Cakewalk and Cubase products and only on the PC. Like everything else in computer music production, you win some and you lose some. Except in this case, players without access to this plug-in maybe losing out big-time. Unless you're a master guitar player with access to a MIDI guitar, I strongly recommend you pick up this little beauty right away.

> * * * * | Rhythm 'n' Chords 2 Musiclab | www.musiclab.com

Todd M. Fav (aka LAX) is an audio consultant and content creator working in and around the games industry. E-mail him at todd@lax-element.com

Sounds like a revolution.





rev-ō-lü-shŭn (noun)—a radical or complete change from the old way in which things were done.

The SoundMAX Game Audio Toolkit gives you a new way to create and control animated audio that makes your game sound *real*.

Available for PC, PlayStation 2, and Xbox developers.

Sony On-Line Entertainment, Infogrames, PitBull, SCEA, and many others have chosen SoundMAX—it's time for *you* to join the revolution!

STACCATO

A Subsidiary of Analog Devices, Inc.



Warren Spector's

Sid Meier!

n the world of game development, Sid Meier is as close as one can get to being a man who needs no introduction. Since co-founding Microprose in 1982, Sid has designed and programmed dozens of games that have been heralded as nothing less than revolutionary, ingenious, and influential to all who follow in his footsteps. Sid lends his experienced hand to Firaxis Games as chairman and creative director of the recently released Civilization III and the upcoming Sid Meier's Sim Golf.

This month's questions were provided by Warren Spector, who has designed numerous critically acclaimed games for Origin Systems, Looking Glass Studios, and Ion Storm. Warren is currently executive producing Ion Storm's upcoming Deux Ex 2 and Thief 3.

Warren Spector. What are Sid Meier's inspirations? Do you play a lot of games? Do you look to the cultural zeitgeist? Do you specifically and consciously look outside the universe of games for fresh insights and ideas?

Sid Meier. Most of my game ideas trace back to my childhood, to things that I became fascinated with at some point during my childhood. Pirates, airplanes, trains, history, and the Civil War were all interests of mine at one time or another.

WS. How do you start the game design process? Do you typically have a moment of gameplay in mind? Or maybe a story or fictional context? Maybe a single game mechanic you think would be cool? A particular fantasy you want to allow players to experience or an overall experience you want them to have? A mood you want to evoke or a message you want to convey? Where does a design start for you?

SM. In starting a design I focus on two key moments. The first time a player starts the game, he or she needs to be quickly drawn into the game. At the end of the game, the player should have a sense of having come a long way since the beginning to a satisfying conclusion and be tempted to play again.

WS. How much documentation do you do before beginning to work? Are you a preplan-as-much-as-possible guy or a prototype-and-revise guy? I've always heard the latter, but I want details!

SM. There's really no preplanning when we start a new game. We build the game using stuff we already know, with the idea that our players will already know this stuff too, and they'll be able to jump right in. Later we do research to add depth, create scenarios, and get the details right, but not until we have a fun game.

WS. How much "real work" do you do these days, and how



Sid Meier: Happiest when programming.

much of your time is spent conveying a vision to a team, or melding various team members' spins on the game into a seamless whole, or just dealing with team and studio management issues?

SM. Actually, I enjoy programming and I don't enjoy management, so I'm generally the lead programmer on at least one project.

WS. How do you explain your success? You've probably worked in a greater variety of genres than anyone else in this business — science fiction, historical sims, pirate games, espionage adventures. Do you think your greatest successes were driven by the appeal of a specific genre or fiction, or were there gameplay differences that made the difference, sales-wise?

SM. I don't really know how to predict the success of a game. In hindsight, it might seem that

doing CIVILIZATION was a no-brainer, but at the time it was a real departure for Microprose. At the time, strategy games were considered boring and complicated. I write games that I think I would like to play and hope there are some other people out there who will like them as well.

WS. How tight is the link between genre and gameplay? In other words, can the same mechanics be applied to a sci-fi game as to a historical sim? Does genre dictate gameplay and game mechanics, or do the mechanics come first and then the genre?

SM. We pick the game topic first and then worry about the mechanics. CIVILIZATION started out as a real-time game and switched to turn-based. PIRATES! was a combination of storytelling, adventure, and action. I tried three different approaches to the DINOSAUR game — turn-based, real-time strategy, and a card game approach — before finally giving up.

WS. A lot of folks in my studio look at some elements of ALPHA CENTAURI in particular as a model for some of the things we hope to do in future immersive simulation games. Do you ever look at other people's games, regardless of genre or game style, and see some of your own ideas embodied in them? Conversely, do you ever ask yourself why more game developers don't adapt your ideas to their own work? In other words, how do you feel your games have influenced the development of games and gaming?

SM. I think there is a continual sharing, borrowing, and building upon game ideas among the design community. As long as each game also introduces some new ideas and innovation, this is one of the strengths of our industry. Certainly the standardization of interfaces and controls has made games easier to play. I still love to play games. I hope other designers will continue to create great games so that I can play them, and occasionally borrow an idea or two.



For the power you need in creating your next 3D game, focus on the high performing line of graphics workstations from Core Microsystems. With more than a decade of experience in professional level workstation technology, you can be sure that Core will deliver the specs you want in a high performance workstation, at unbeatable

For lightning fast 3D rendering, each system combines the latest high-speed processing power with industrial strength accelerators such as the Oxygen and Wildcat series from 3D Labs as well as the nVidia based accelerators from Elsa. With open GL functionality and up to 256MB virtual memory these new-generation graphics accelerators deliver outstanding performance to leading 3D applications.

And to top it off, Your 3D package is backed by an unbeatable 3 year parts and labor warranty, with full first year on-site service included. For the ultimate combination of power, price and performance, call Core to maximize your creative potential.

Call today for a free quote or visit our website at http://www.coremicro.com.

Order direct by calling 1-800-886-2752.

PhEnix Dual P4 Xeon

- Dual 2.0 GHz Intel P4 Xeon Processors
- · Mainboard with Intel 860 chipset
- 400MHz front side bus
- 512 MB Rambus system memory
- · Onboard LAN, dual channel U3 SCSI
- IBM 18 GB U3 SCSI Hard drive, 10,000RPM
- 24x10x40 CD R/W
- 3D Labs Wildcat 5110 Accelerator/144 MB
- Windows 2000 Professional or Windows XP
- Full Tower Case with 460 Watt P/S
 - \$5740

PhEnix Dual AMD-Athlon

- Dual AMD 1.6 GHz processors · Mainboard with AMD 760 chipset
- 266 MHz frontside bus
- 512 MB DDR PC-2100 system memory
- Onboard LAN, dual channel U3 SCSI
- IBM 18 GB U3 SCSI Hard drive, 10,000RPM
- 24x10x40 CD R/W
- 3D Labs Wildcat 5110 Accelerator/144 MB
- Windows 2000 Professional or Windows XP
- Full Tower Case with 460 Watt P/S
 - \$4395

PhEnix Solo-P4

- 2.0 GHz Intel Pentium 4 Processor
- Intel Mainboard with 850 chipset
- 400MHz front side bus
- IBM 18 GB U3 SCSI Hard drive, 10,000RPM
- 512 MB Rambus system memory
- 24x10x40 CD R/W
- Elsa Gloria III Accelerator/64 MB
- Windows 2000 Professional or Windows XP
- . Mid Tower Case with 250 Watt P/S
 - \$3275

All prices and specifications subject to change without notification.

Graphic created using LightWave by Epic Software Group, The Woodlands, TX www.epicsoftware.com





Mathematical **Growing Pains**

his month, I'm going to talk about how to represent lines and planes, the sorts of linear entities that programmers manipulate all the time.

In high school I was taught that the equation y = mx + b is a groovy way to represent a line in 2D. The equation is useful because m represents the slope and b is the y-intercept — that is, the line intersects the y-axis at (0, b). This representation is good if you don't have a lot of higher math experience and you just want to draw a line on a piece of graph paper: b gives you a starting point, and m gives you the direction to go from there.

Years passed, until one day I was programming some pretty advanced 2D games; by then I had used y = mx + b for visualization so often that I thought of it as the primary way to talk about lines. So I tried to make systems that represented lines with two floating-point numbers, m and b.

But what happens when a line is vertical? Its slope is undefined. In that situation, in high school, you'd just write x = k, which seemed simple enough. But with games, you have to think about more complex situations, like lines that are smoothly rotating from frame to frame. And you're writing code that uses limited-precision numbers, so your computations become numerically illconditioned when the lines are steep, because m is such a huge number. To fix this problem, you put a bunch of if statements into your code to change the computation based on what neighborhood the slope is in. That's not desirable from a software engineering standpoint, and the computational discontinuities

(these happen as your parameters cross from one if scenario to another) may cause subtle but disturbing things to occur. See the pseudocode in Listing 1 for an example.

These problems go away when you make a simple mental adjustment and use ax + by + c = 0 as your line equation. This is like the slope-intercept equation, but before a division has taken place: if you divide ax + by + c = 0 by b (the bfrom this equation, not the b we were talking about before), you get the slopeintercept form. The slope and intercept shoot toward infinity when b is near 0, meaning the line is vertical. So ax + by +c = 0 is more robust because it doesn't divide by b.

As a bonus, the surface normal of the line is (a, b), and the distance from the line to the origin is c. You can easily read these features out of the equation. and being a game developer, you're more likely to care about these things than the y-intercept. Though we now need three floating-point numbers to talk about our line, a, b, and c, that extra number buys convenience and software reliability. The software becomes more reliable because the precision of our computations is more isotropic. In other words, it doesn't matter so much what direction the line goes in.

To sum up, my learning of y = mx + b

as the way to talk about lines had impacted my effectiveness in making games; the alternate representation removed those blockades.

Extending Lines to 3D

fter a while, I'd made enough 2D games and decided to try 3D. When I first tried to formulate line equations in 3D, I got confused. In 2D, ax + by + c = 0 had been the best thing since sliced bread, so clearly I wanted to extend that equation to 3D. The obvious candidate is ax + by + cz + d = 0. I knew from reading books that this was the equation for a plane. Extending my line equation to 3D requires adding z in somehow, right? How else could I possibly add a z that would make any sense?

The problem is that ax + by + c = 0, which I'd thought was an enlightened way of representing a line, is not a line equation at all — and neither is y = mx+ b, for that matter. It's a plane equation, and it only worked because lines and hyperplanes in 2D are the same thing (where my temporary definition of a hyperplane is "that which divides space into two halves").

There is an equation that works for all lines regardless of the space's dimension. It is $L = p_0 + \nu t$, where L represents the set of points comprising the line, p_0 is an arbitrary point known to lie on the



JONATHAN BLOW I Jonathan is a game technology consultant living in San Francisco. Film that influenced this article: Mulholland Drive, the GREATEST FILM EVER. They can pretty much stop making movies now. It's time to work on games. Jon's e-mail address is jon@bolt-action.com.

line, v is the direction vector that the line travels in, and t is the time parameter. When we get used to thinking about lines this way, we build up intuition that is valid no matter how many dimensions we're dealing with. We say that this is the parametric form of the line, as varying the parameter t will give you every point in t. If t is the dimensionality of your space, then this equation requires t numbers' worth of storage if you're being lackadaisical, or t 1 if you're being hardcore.

Simultaneous Equations?

S o why is ax + by + etc. the equation of a hyperplane and not a line? It's because it takes n degrees of freedom (represented by the coordinate variables x, y, . . .) and, by binding them together with the equal sign, places one constraint on that system of variables. This linear constraint removes one dimension; it flattens the space in the direction of the gradient of the equation (this gradient is the same thing as the normal of the hyperplane). The resulting space has n-1 dimensions: in 2D, you get a line; in 3D, a plane; and in 4D, you get a 3D hyperplane.

Suppose we didn't want to use the parametric form for a line in n dimensions. Instead, we could represent the line by starting with the full *n*-dimensional space and squashing it n-1 times, because n - (n - 1) is 1, the dimensionality of a line. We can do this using n-1linear equations simultaneously. Simultaneous linear equations are the same thing as a matrix. So we're storing an n by n-1 matrix, which uses a lot of storage space, and furthermore, it's not guaranteed to behave nicely. Suppose two of our equations try to squish the space in the same direction. After the first equation acts, there's nothing left for the second one to do; so the second equation doesn't reduce the space by a dimension (in fact, it leaves it unchanged). After all our n-1 squashings, the remaining entity will have one more dimension than we expected; instead of a line, it will be a 2D plane.

LISTING 1. An example of how a singularity in mathematical representation affects code.

```
struct Line {
    float slope, y_intercept; // 'slope' == m, 'y_intercept' == b
    bool is vertical;
                              // or else declare this, meaning the above are invalid
                              // used only if the line is vertical.
    float x_value;
};
bool intersect_with_vertical_line(Line *vertical, Line *non_vertical, float *x_result,
float *y_result) {
     *x_result = vertical->x_value;
     *v result = non_vertical->slope * vertical->x_value + non_vertical->y_intercept;
     return true;
bool intersect_nonvertical_lines(Line *line_1, Line *line_2, float *x_result, float
*y_result) {
     // Hope this denominator is not small.
     *x result = (line 2->slope - line_1->slope) / (line_2->y_intercept - line_1-
     // Choice of line_1 below is arbitrary, hope we're well-conditioned.
     *y_result = line_1->slope * (*x_result) + line_1->y_intercept;
     return true;
bool intersect_lines(Line *line_1, Line *line_2, float *x_result, float *y_result) {
    if (line 1->is_vertical) {
         if (line_2->is_vertical) return false;
        return intersect_with_vertical_line(line_1, line_2, x_result, y_result);
    if (line_2->is_vertical) {
         return intersect_with_vertical_line(line_2, line_1, x_result, y_result);
    return intersect_nonvertical_lines(line_1, line_2, x_result, y_result);
```

We then need to break out some advanced linear algebra to deal with the situation. Naive game programmer code, just consisting of a big hand-derived vector equation worked out on paper, will end up dividing by 0 somewhere and freaking out. More experienced programmers might use a matrix equation, but black-box matrix methods get screwy too; we end up with a situation where the determinant of a matrix is 0 and we want to invert it. The matrix has no inverse. Badly written code tries to invert it anyway, and thus produces inaccurate results or NaNs. Better matrix code takes stock of the situation with an if statement and, if the determinant is within

some epsilon of 0, reduces the dimensionality of the matrix and solves a reduced-dimension problem. But picking suitable epsilons is not easy, and numerical discontinuities are introduced by the if statement.

All this should sound familiar from an engineering standpoint — it's the kind of thing we were doing with y = mx + b when the line became vertical, and all the same problems arise. Cases of determinant 0 are often called "degenerate," but I think they are quite natural and inability to deal with them indicates weak methodology.

Imagine that you have three different planes, all passing through the origin,

rotating freely in 3D. You want to find the intersection of those planes. Most of the time, they intersect in a point; but if two of the planes coincide, then all three intersect in a line; and if all three coincide, the answer is a plane.

To solve this intersection problem using beginner's linear algebra, we write a matrix equation $p = A^{-1}d$ that finds the solution; but hard-coded into this equation is the assumption that the answer is a point. When the answer is not a point, A has determinant 0, so the equation is unsolvable. But what's the big deal? Sometimes planes are coplanar, just like sometimes lines are vertical. Why should that be a problem? The problem goes away when we stop treating matrices as black boxes that we want to invert, and instead start decomposing them and looking at their intrinsic properties. The QR and singular value decompositions become useful to us at this point.

Common Mathematical Misconceptions

started this article with the question of how to represent a line. As 3D programmers we get past these problems early on, if only because we can't do lines in 3D otherwise. About the varying representations of a line, I want to develop an analogy: they are like other concepts that we work with from day to day, rooted in the core of our thinking, that are misleading in 3D and don't even work in higher dimensions. I'll now describe the biggest ones, the axis of rotation and the cross product.

The Axis of Rotation

hen learning 3D math, once we get past the inconvenience of Euler angles, we find that all rotations can be represented by an axis vector, about which we rotate, and an angle, representing the magnitude of the rotation. Perhaps we visualize a rotation as a wheel turning on an oriented axle.

The problem is that the whole concept of "axis of rotation" only works in 3D.

In 2D, rotations occur around a central point, and maybe we think of a nonexistent axis sticking out of the plane to help us visualize this. But a much more reasonable way to think of rotations is to speak of the "plane of rotation" rather than the axis. In n dimensions, any rotation occurs within a two-dimensional plane, and the object rotates "around" however many dimensions are left in the space. In 2D space, you rotate around a zero-dimensional subspace, a central point. In 3D, you rotate around a one-dimensional vector subspace. In 4D, you rotate around a two-dimensional planar subspace. (In 3D, the surface normal of the plane of rotation is the axis vector we are used to thinking about. In higher dimensions, using this definition of rotation, it's no longer true that you can reproduce an arbitrary orientation with only one rotation.)

I tend to think of rotation as "the thing that binds together any two dimensions of our space." In 3D, we have three canonical planes of rotation: the *xy* plane, which binds things that leave *x* to entering *y*, and likewise for *yz* and *xz* planes. Any rotation occurs within a 2D plane that is a linear combination of these three canonical planes. In 4D, there are six such canonical planes.

David Hestenes (see For More Information) uses a different terminology from what I use here; he speaks of "simple rotations," which occur within two planes, and "arbitrary rotations," which can reproduce any orientation in the space. I find this terminology unappealing, since in some higher-dimension spaces, such as 4D, a non-simple "rotation" may have no eigenspace — that is, no "axis" of nonzero dimension around which the rotation pivots. So I find it hard to visualize the thing he calls a "rotation." But your mileage may vary.

The Cross Product

The cross product is a fundamental piece of 3D math that we use all the time. But we were taught incorrectly what the cross product is and how it works, with the result that we often use it

improperly, in subtle ways.

We are usually taught only about the cross product in 3D. But what is the cross product in four dimensions and higher? Does the concept even make sense? Because two linearly independent vectors determine a 2D plane, it is possible for us to interpret the results of the cross product in *n* dimensions as the subspaces we were rotating around just a few paragraphs ago: in 2D, the result is a scalar; in 3D, a vector; and in 4D, a 2D-planar thing.

Following this scheme, when we take the cross product in 3D, we think of it as returning a vector result. Unfortunately, this result is wrong, and we see this in a few places. A prominent symptom is that "surface normal vectors" can't be transformed in the same way that plain vanilla vectors can; if you are transforming vectors by some transformation T, you need to transform normals by $(T')^{-1}$. Beginning 3D programmers may not run into this problem, because if T is just a rotation, its inverse is equal to its transpose: $(T')^{-1} = T$.

This difference in transformations is necessary because the cross product is weird. We are providing two vectors as arguments of the cross product, and those vectors determine a 2-plane if they are not colinear. But the cross product implicitly returns to us the dual of that plane, its normal vector. So we think we're talking about a vector, but we're really talking about a plane through the origin. The plane occupies whichever two dimensions its normal vector does not: because of this, transformations can affect the plane in ways that we would not see if we considered its normal vector in isolation.

To ensure that we always pick the right transformation, we can say that the output of the cross product is a thing called a form, which one might think of as a transposed vector. The form interacts with matrices in all the ways you'd expect a row vector to behave. Smart physicists have been dealing with the differences between point-like and plane-like things for a long time; eventually someone invented Einstein index notation, which

CREATIVE JUNGLEGYM

If your work looks like recess to everybody else, check out the Compaq Evo™ Workstation W8000. Equipped with the latest Intel® Xeon™ Processor technology, it's designed and optimized for intensive graphics and video applications. Giving you the freedom to explore the widest range of creative possibilities any right-brain-heavy graphics mind can dream up. And, with a broad array of software partners, Compaq offers some of the best Digital Content Creation solutions. But hey, you can let the left-brain guys worry about that.

to find out more go to compaq.com/workstations or call 1•800•888•0414



The Power of MACROPROCESSING

Compaq PCs use genuine Microsoft® Windows® www.microsoft.com/piracy/howtotell

Compaq is not liable for editorial, pictorial or typographical errors in this advertisement. Compaq and the Compaq logo are registered in the U.S. Patent and Trademark Office. Evo is a trademark of Compaq Information Technologies Group, L.P. in the U.S. and other countries. Intel, the Intel Inside logo, and Intel Xeon are trademarks or registered trademarks of Intel Corporation or its subdiairies in the United States and other countries. Products and company names mentioned herein may be trademarks and/or registered trademarks of other countries. © 2001 Compaq Computer Corporation.

COMPAQInspiration Technology

Evo W8000

COMPAG

helps demystify things. Jim Blinn (see For More Information) wrote two articles that discuss the Einstein notation from a graphics programmer's point of view. But this whole tensor algebra approach gets pretty complicated, so some new-school physicists are evangelizing Clifford algebra (also known as geometric algebra) as a method of simplification.

Clifford algebra defines the "wedge product" of two vectors in a way that is similar to the cross product, but it returns a nonvector result; that result is a plane-like thing called a bivector. You can take the wedge product of a bivector and another vector to get a volumetric this algorithm, which only takes mesh geometry into account, operates on 3D vectors; it uses 3D plane equations that are derived and evaluated using the cross product and the dot product. But to take vertex color and texture coordinates into account, we need to generalize the algorithm to higher dimensions.

We hit a wall when we try to move the algorithm to higher dimensions, because each face of our mesh imposes a two-dimensional constraint on the quadric error metric. When we're in three dimensions, this constraint can be represented as the hyperplane ax + by + cz + d = 0, which we're used to playing with. But

When we get used to thinking about lines and planes parametrically, we build up intuition that is valid no matter how many dimensions we're dealing with.

trivector, and so on. The Clifford product of two vectors gives you a result containing both scalar and bivector parts; it is the dot product and cross product all wrapped together. This unification enables us to do things that make life easier, like dividing an equation by a vector or a plane.

In some references, the 2D version of the cross product is called the "perp-dot product" (see F. S. Hill's Graphics Gem in For More Information). Pertti Lounesto's book describes higher-dimension cross products that are different from the one I've mentioned here.

Why We Should Care About N-Dimensional Generality

Recently, to generate levels of detail for humanoid character meshes, I was implementing Garland-Heckbert Quadric Error Simplification (see For More Information). The basic version of

when we go up to five dimensions (three spatial dimensions plus two texture coordinates per vertex), we no longer have such a tidy hyperplane equation to represent what's going on. Each face of the mesh defines a 2D plane, but now a 2D plane is just a small strand in the 5D space, so we need to represent it parametrically. This is exactly analogous to the way ax + by + c = 0 stopped working for lines when we jumped from 2D to 3D.

Another way of looking at the problem is this: in 3D we usually get a plane from two vectors by taking the cross product. But if we're not conversant in advanced linear algebra, it is unclear how to perform this process in 5D. So be sure to eat your multi-dimensional Wheaties.

In their paper, when the time comes to elevate above three dimensions, Garland and Heckbert shift gears away from the hyperplane approach and re-derive their algorithm differently. But if you start with an all-encompassing approach (such as Clifford algebra) from the beginning, the

FOR MORE INFORMATION

Blinn, Jim. Jim Blinn's Corner: Dirty Pixels. Morgan Kaufmann, 1998.

Dorst, Leo. "GABLE: A Matlab Geometric Algebra Tutorial." http://carol.wins.uva.nl/~leo/clifford/ gable.html

Garland, Michael, and Paul S. Heckbert.
"Simplifying Surfaces with Color and
Texture Using Quadric Error Metrics."

Proceedings of IEEE Visualization, 1998.
http://graphics.cs.uiuc.edu/~garland

Hestenes, David, and Garret Sobcyk.

Clifford Algebra to Geometric Calculus:

A Unified Language for Mathematics
and Physics. Kluwer Academic
Publishers, 1987.

Hill, Jr., F. S. "The Pleasures of 'Perp Dot' Products," in *Graphics Gems IV*, ed. Paul Heckbert. Academic Press, 1994.

Lounesto, Pertti. "Clifford Algebras and Spinors." London Mathematical Society Lecture Note Series #239. Cambridge University Press, 1997.

algorithm works no matter what dimension you deal with, and you never have to get confused or change your mode of thought. You also end up with a shorter derivation than that used in the Garland-Heckbert paper.

So the traditional tools of 3D vector math definitely hinder us in these kinds of pursuits, and broader approaches can help us. I must emphasize that Garland-Heckbert is not an obscure algorithm; it's one of the best, simplest, and most widely used methods of performing mesh simplification.

ACKNOWLEDGEMENTS

Thanks to Chris Hecker for redirecting the overly negative energy that originally permeated this article concept, and for some pointers regarding matrix decomposition.





CALL FOR NOMINATIONS

Presented by the International Game Developers Association, the Game Developers Choice Awards are game development's most prized honors. All professional game developers are eligible to nominate and vote for the developers who transcended the state of the art in 2001.

"...this award, voted on by the game community, is easily the most important in my mind"

- Will Wright, Maxis

Awards will be announced at the Thursday, March 21 ceremony at the Game Developers Conference 2002 in San Jose, California.

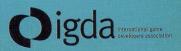
www.igda.org/awards

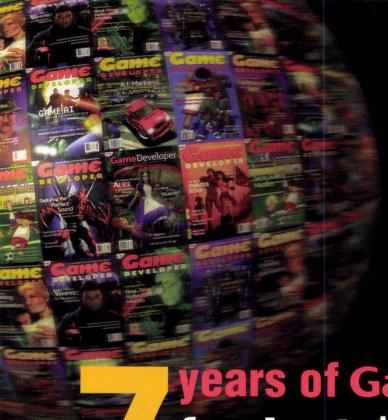
Nominations open January 2, 2002

Nominations close February 1, 2002 Sponsored by:



Presented by:





years of Game Developer for just \$149

Each CD-ROM contains:

- ✓ Full text of feature articles, columns, and product reviews
- ✓ Articles, artwork, and diagrams exactly as they were printed
- ✓ All code and applications that accompanied the articles

Great content. Great prices!

NEW! ✓ May 2000 to April 2001 \$49

✓ May 1999 to April 2000 \$39

✓ 1994 to April 1999 \$99

Or get everything:

✓ Premier issue to April 2001 \$149

Order by phone:

800.444.4881 (US) or 785.841.1631 (international)

Order online:

www.gamasutra.com/store

Sponsored by:



Making Trees Work

ow many times as an artist have you wished that you were working on Tetris? Those small, colored blocks, slotting together — simple, Spartan, square. It



FIGURE 1. Formula 1 cars through the ages.

may not be creative nirvana, but at least you know where you are with a square — four sides, four corners; you can't really go wrong. Other shapes are somewhat less forgiving, and those that relate to the real world are usually amongst the most awkward.

Those Were the Days

s we rocketed out of the 1970s and through the 1980s, toward the 16-bit wonders of the early 1990s, we left

the minimalist angularity of PONG and the harsh, blocky SPACE INVADERS behind us, heading for the world of graphical beauty that glimmered on the horizon. Videogames slowly made the transition from the symbolic, where the sports car in our racing game looked like it had been drawn by my three-year-old brother, to the representational, where a Ferrari had at least a passing resemblance to the vehicle it was portraying. Figure 1 shows three transformations videogame cars have gone through over the ages.

Suddenly, games were no longer judged simply on how much fun they were; they also had to look fantastic. Teenagers across the globe found themselves huddling together around a single copy of a magazine they'd smuggled into school, breathing heavily, and letting out the occasional sigh of rapture. This time, however, it wasn't Marlena (21, enjoys skiing and topless volleyball) stoking the fires of their desire, it was a game. The centerfold had stepped aside; the pinup had given way to the pixel.

As with most things technological, the increase in graphical quality continued to grow exponentially. A palette of 256 colors soon became one of over 65,000. Screen resolutions grew and animation broke free from the confines of a simple loop. And then 3D hit us. Like a stampeding herd of hyperactive rhinoceroses, the 3D revolution trampled the sprite

and all its flat friends under its huge, polygonal feet. The irresistible bulk of an industry ready to move on, backed by the widespread availability of the hardware necessary to do the job at an affordable price, moved games and their graphics to the next level.

And here we are. By the time I finish writing this article, there may well have been another wave of graphics cards released, capable of real-time volumetric refraction with a dedicated subprocessor specifically created to deal with dynamic nasal hair deformation. By the time my cup of coffee gets cold, the next generation of consoles may well be on the shelf, promising to draw things so fast that the only way to prevent your TV from melting is to immerse it periodically in liquid nitrogen. Today's game players want more than a yellow circle with a mouth being chased around a maze by ghosts. They want their games to be set in a world they can believe in.

Whether a game is trying to re-create a location that is real or it takes place in a world of fantasy, more often than not certain elements of the environment occur in both. Rocks are usually rocks, whether they're on an ancient Egyptian battlefield or in the lair of the mighty demon Aarath'ak the Unhelpful. Whether you find yourself snowboarding in the Rockies, or racing through the valleys of the planet Sprag-Thurman VIII, a tree



HAYDEN DUVALL I Hayden started work in 1987, creating airbrushed artwork for the games industry. Over the next eight years, Hayden continued as a freelance artist and lectured in psychology at Perth College in Scotland. Hayden now lives in Bristol, England, with his wife, Leah, and their four children, where he is lead artist at Confounding Factor.

(excepting certain localizing factors) is basically a tree.

I Hate Trees

rom the outset, we have a few problems when it comes to trees. First, just like most things that nature creates for us, trees were never designed to be modeled economically in three dimensions. The whole branching concept is





FIGURE 2. Two different species of tree.

death to even the most robust polygon budget, and that's before we get to the leaves. Second, besides the fact that each specific species of tree is distinct from every other species, every single tree is always, in itself, unique. And finally, a tree is very rarely found in isolation; more often than not it will be in a group, and these groups (clump, copse, woods, or forest), are going to be outside, which is usually quite a large place to fill. Figure 2 shows the differences between two different species of tree.

There is a clever solution to this problem: simply set your game in a submarine. There are no trees in submarines. Or how about space? A space combat game will remove any need for trees. Just as long as you don't land on a deciduous planet, that is. However, chances are that neither of these scenarios will fit with your current project, and so a strategy to deal with trees will definitely come in handy.

Context

To begin with, it is worth looking at how trees fit in with your game design. This may seem like overkill, but it will ultimately affect the choices you make about the best way to approach building your trees.

As is often the case, some of the first questions to ask have to be: What kind of world are we building? Will players have full access to the whole landscape, or will they be limited to certain predefined areas? How much space will a level represent, and what percentage of this is likely to be exterior space? Will trees generally be used to beautify the background during bouts of tightly scripted action, or will they most likely be used to break the monotony of large, open plains? Do trees have a more important role within the gameplay? Will their placement be carefully integrated with the level design as cover to be used by players when they creep around, stealthily to avoid detection by the enemy? Do they provide a hiding place from which the Giant Spleen-Beast of Gaarg will charge at the end of the level?

Obviously, approaching any element of a game's visuals in isolation from its context is like painting a portrait over the telephone. Chances are, you're going to have to do it again. In this respect, the items of primary concern are: Are the trees themselves likely to be under close scrutiny by the player? What level of detail will strike the best balance between quality and speed?

It is entirely possible that your engine is able to apply some form of dynamic level-of-detail adaptation to geometry, which gives you extra breathing space when it comes to limiting your model's complexity. However, regardless of what hardware manufacturers around the world would have us believe, you and I both know that unless your engine draws its power from the Dark Prince himself, economical design is what we're after.

What Makes a Tree?

Before we set about building one, it's worth having a quick look at what makes a tree. Chances are, you're pretty familiar with the basic tree. Bear with me, however, while I break it down into a few basic elements:

Roots. Depending on the species,

roots are generally underground, but particularly with larger, older trees, roots have a significant presence above ground.

Trunk. This fairly straightforward central mass of the tree typically starts fat, ends thin.

Branches. Mainly quite random, but some species, such as firs, have surprisingly uniform (fractal) branching structures.

Bark. There is enormous diversity from species to species.

Leaves/needles. Again, these are hugely diverse in shape, as well as relative size, when compared to the parent plant.

So that, more or less, is a tree. No problem. Couldn't be easier. Familiarity with the subject when trying to re-create it, whether in paint, plaster, or polygons, is bound to make the process easier, isn't it? Unfortunately, a faithful recreation of something as complex as a tree will, by the time you've scattered trees across your level, most likely have your engine cowering in a corner, begging for mercy. In this instance, as is often the case when considering graphics for a game, achieving realism will involve a certain element of stylization and a level of approximation that fits within the restrictions of your particular project.

Cheap and Cheerful

Before you create a single vertex, it is helpful to divide the trees up into two groups: generic, or filler trees, and those that will have a more prominent place within a level, the feature trees (for want of a better name). A filler tree, as the name suggests, is most likely used in numbers to break up empty background space, and to create a more attractive, detailed exterior. The feature tree is generally the setting for some important action, whether gameplay or cutscene, and as such the player spends more time looking at it. Because the feature tree is a more central part of any scene you are creating, its level of detail, while dependent on engine limitations, is considerably higher than that of the filler tree.

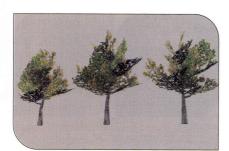


FIGURE 3. Asymmetrical trees with rotation.

It is unlikely that you have either the time or the resources to create a vast array of individual trees that mimic their uniqueness in the real world. It is important, then, to design your filler trees to work well as a set that can be distributed as well as possible to create the illusion of variety. Scale variations and rotation (as long as the trees are not symmetrical) can be mixed with vertex coloring as well as light and shadow variations across a landscape to increase the apparent number of different trees that have been created. Figure 3 shows three asymmetrical trees with rotation.

Once you've chosen your basic species of tree (and here I mean, fir, palm, deciduous, and such, not *Chamaecyparis lawsonia*), you can create the object that will form the trunk. For a filler tree, this should have as simple a shape as possible (a triangular cross-section should be perfectly adequate in most cases), with the number of branches kept to a minimum and no secondary branches needed. Most of the work for this kind of tree can be done in texture. Detailed branch and leaf structures can only be represented economically in texture, and this is especially true in the case of the filler tree.

Foliage

The extreme economy of the past, with two crossed polygons displaying a complete tree in texture, may be some way behind us. However, using a combination of solid geometry for the trunk and major branches and some crossed planes representing the foliage can provide an adequate compromise.

Creating a useful texture for this part of the tree is of particular importance, and the following are four methods that are worth considering.

Hand Painting

and painting is useful if your visuals are adopting a particular style,



FIGURE 4. Van Gogh's unhelpful poplars.

and it's especially worth considering if you plan to work at a resolution of 128 dpi or lower. You can create leaves and branches in most decent paint packages



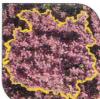






FIGURE 5. Photo to texture (stages one through four, from left to right).

and use features such as cloning and image hoses to distribute elements such as leaves in a variety of ways. Benefits of this method include the relative ease with which an alpha channel can be created, as well as the level of control the artist has over the exact positioning of the features within the texture.

Scanned Illustrations

ften overlooked as unsuitable source material, paintings and illustrations of trees can be a very useful starting point for textures. O.K., if you choose Van Gogh's Poplars on a Hill (Figure 4) as inspiration, you may find yourself struggling to make use of his mad wavy lines, but there are some examples that are more friendly. Illustrations of different tree species, found in botanical encyclopedias for example, are often highly detailed and accurate. The advantage of this kind of image is that lighting is often very diffuse, without the hard shadows that can make photographs unusable. Also, the foliage is generally on a white background, which makes it easier to separate out than a busy photo.

Photographs

scanned or digital photo is easily the most common source of textures, and perhaps the most difficult part of the process of turning a photo into a useful foliage texture (see Figure 5) is generating an alpha channel. A solid mass of leaves will be of limited effect when trying to give the impression of complex branching, and so it is important to use transparency within the texture to create the shapes that are

far too costly to build into the geometry.
Supposing you have a fairly standard photo of a leafy canopy, the question

then becomes how best to select only the areas that are to be left visible, so that the others can be masked out. I may be wrong, but Photoshop doesn't yet come with a "choose a good selection of leaves" tool. The problem is that most selection methods can't discriminate between the leaves and branches that you want to keep and those which will leave the texture too cluttered.

Unfortunately, as with so many things in life, for a completely satisfying result, you'll have to do the job yourself. Selecting by color and altering the fuzziness, or creating a threshold layer and adjusting the levels, may provide a rough starting point for selection. Still, you'll have to refine these characteristics manually.

Rendered Images

useful addition to the essentially two-dimensional source material mentioned thus far is the rendered tree. Most major packages have a selection of tree-generating programs with names like "Tree Master Plus" and "Auto Foliage 500." While each is different, they all produce similar results that are too detailed to be used as ingame geometry but can be extremely useful for creating textures. Perhaps the most appealing aspect of this approach is the automatic generation of an alpha channel when the image is rendered, which both saves time and increases accuracy (see Figure 6).

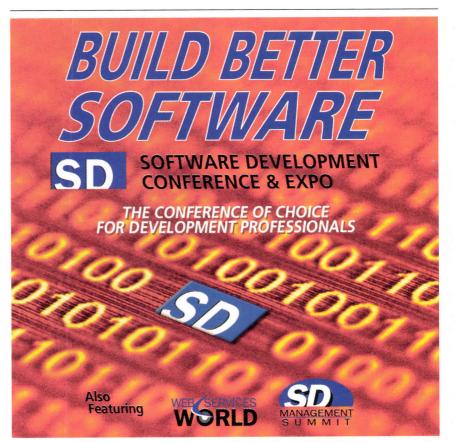
Whichever method you use to create the textures, assembling the geometry on which they will be mapped in order to create the appearance of a tree's canopy must be done carefully and take into consideration the way in which the trees will be encountered. If, for instance, the trees will be viewed from beneath, the orientation of the geometry containing the branch and leaf textures needs to take this into account. The same holds true for the trunk geometry; extra detail can be worthwhile around the lower trunk and root region if players are likely to spend their time in this area.

In the end, finding the best way to make trees is a bit like finding the best way to bake bread. There are a million different recipes. Clever use of texture, coupled with economical modeling, can be the best compromise between detail and practicality, and while the process may be awkward, getting the right result is certainly well worth the effort.





FIGURE 6. Render plus alpha channel equals texture.



5 FULL DAYS of world class instruction, exhibition and special events!

KEYNOTES:	▲ Anders Hejlsberg ▲ Steve McConnell ▲ Bruce Eckel ▲ Elliotte Rusty Harold ▲ Scott Meyers ▲ Larry Augustin		▲ Clay Shirky ▲ Karl Wiegers	▲ Richard Hale Shaw ▲ Robert C. Martin ▲ Johanna Rothman ▲ Scott Ambler	
SPEAKERS INCLUDE:			▲ Bertrand Meyer ▲ Karl Wiegers ▲ Martin Fowler ▲ Dan Saks		
ADVANCED CURRICULUM IN:	▲ JAVA				
			RY 1ST & SA		

Experience games that break genres, push the boundaries of gameplay and redefine the interactive experience.

The 4th Annual Independent Games Festival

The 4th Annual Idependent Games Festival

at the Game Developers Conference

March 21-23, 2002 · San Jose, California

Finalists for the IGF will be announced December 10, 2001.

www.igf.com

Sponsored by:

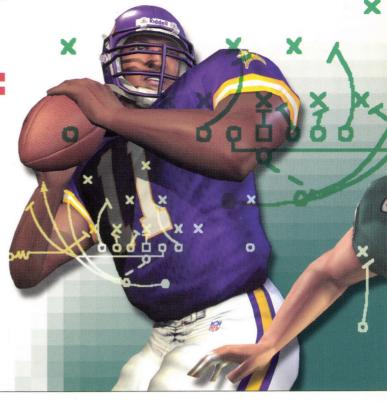
Microsoft DIRECTX®



Members of the Nexon team, developers of SHATTERED GALAXY, recipients of four awards at the 2001 IGF



Play by Play: **Effective Memory** Management



BRIAN HIXON | Brian has a B.S. in math/computer science from Carnegie Mellon University. After losing his mind from playing too many computer games, Tiburon took pity on him and gave him a job in 1998. Brian is a lead programmer for MADDEN PS2, Gamecube, and Xbox.

DANIEL MARTIN | Daniel has a B.S. and an M.S. in computer sciences. After wasting years of his life outside the gaming industry, he saw the light and joined Tiburon in 1998. Today, Daniel is a lead programmer for MADDEN PS2, Gamecube, and Xbox.

ROB MOORE | Rob has a B.S. in electrical engineering. After working on graphics chips and APIs for the Nintendo 64 and Gamecube, Rob decided to see why game developers were always griping at hardware guys. He joined Tiburon in 2000 and now it all makes sense; currently he is Tiburon's chief technical officer. Send comments on this article to MMarticle@tiburon.com

GREG SCHAEFER | Greg has a B.S. and an M.S. in computer science. Greg spent many years working on telecommunication and network applications prior to joining Tiburon in 1998. He now leads MADDEN PC network development.

RICHARD WIFALL | Richard has a B.S. in electrical engineering. He got started in the game industry on 16-bit consoles.

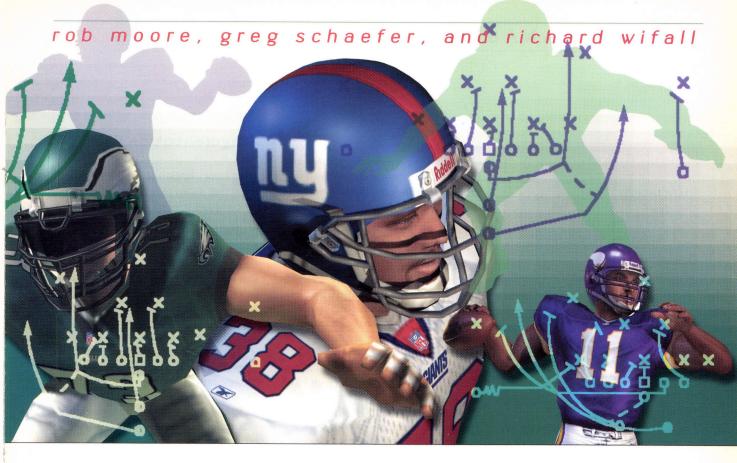
ack before Al Gore invented the Internet, back when 64KB was more memory than any computer would ever need, there was a time when memory managers didn't exist. But gradually, new computer systems came out with larger amounts of memory (Figure 1). Game designers discovered ways to eat up RAM faster than any system could dish it out. It became obvious that managing all this data with a simple static memory map was too much work. Programmers needed a way to create memory maps that could change dynamically at run time depending on the state of the program. Thus, the memory manager was born.

For an excellent introduction to how a basic memory manager works, check out www.memorymanagement.org/articles/ begin.html. This article will assume this basic knowledge and focus instead on things to think about when writing your own memory manager and point out some of the problems you may encounter in using a memory manager. This discussion is based on our experiences in writing and rewriting the memory manager for MADDEN NFL 97 to MADDEN NFL 2002. This article is also slanted toward the C language as written for console game machines. The same principles would apply to C++ and Windows programming, but the details are likely different.

Growing Your Own

S o you know that memory managers are great, but what's wrong with just using malloc and free? They are part of the standard C library, so they should be the best, right? One major problem is that you don't really know what they are doing. If you write your own memory manager, you can generate statistics, add debug code, and add advanced features that might not be found in the default memory manager - features like handles, heap compaction, and virtual memory. The ultimate weapon of game programmers is context. You know the needs of your game and can tailor the memory manager accordingly.

Developing a successful memory manager requires addressing five different issues: ease of use, performance, memory overhead, fragmentation control, and debugging support. Unfortunately, these attributes are all interconnected, and optimizing one can result in substandard results for another. The memory manager designer must govern a delicate balancing act.



At a lower level, memory manager design also requires paying attention to platform-specific requirements. In addition, it may be possible to utilize hardware support to assist the memory manager.

Ease of Use

ith respect to a memory manager, the ease of use consideration really comes down to a single question: Should the memory manager support pointers, handles, or both? When designing a memory manager and dealing with the problem of fragmentation, the use of handles can be very appealing. Unfortunately, while handles provide a straightforward solution to the fragmentation problem, they are much more difficult to use than pointers. A memory manager that supports only handles is essentially pushing its internal complexity back onto the user.

While supporting both handles and pointers is possible, the resulting memory manager is more complicated than one that supports a single method. MADDEN used to support both handles and pointers until an analysis showed that pointers were being used 99 percent of the time. Not surprisingly, when given a choice, programmers used pointers, since they

were the easiest solution. Therefore, we simplified the latest MADDEN memory manager by removing handle support and optimizing the pointer support.

Performance

erformance must be addressed both in terms of speed and consistency. A memory manager that is fast most of the time but slows down dramatically at unpredictable times is unacceptable in a gaming environment. Therefore it is important to understand the issues that contribute to performance. From the memory manager user's point of view,

two operations will impact the game: allocations and recycling.

Allocation performance is determined by allocation policy, free list management, and the use of garbage collection. The most popular allocation policies are best fit, first fit, and

next fit. By organizing the free blocks as a linked list, best fit has consistent O(n)performance, while first fit and next fit have worst-case O(n) and on average O(n/2). By organizing the free blocks as a size-sorted binary tree, best fit has worst-case O(n) and on average $O(n \log n)$ n). By organizing the free blocks as a balanced binary tree, best fit has consistent $O(n \log n)$. Garbage collection applies only to memory managers with handle support, and generally involves a fairly significant performance penalty when it occurs, as it essentially makes a copy of the entire memory heap during compaction.

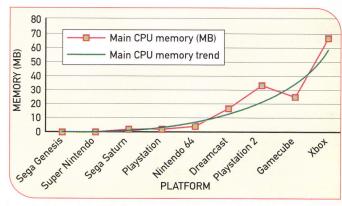


FIGURE 1. Main memory growth for console game machines.

STATIC vs. DYNAMIC MEMORY

STATIC MEMORY

Pros. Fast

Cons. Difficult to maintain

Memory can't be easily reused

DYNAMIC MEMORY

Pros. Objects can be designed to only use memory resources when the object exists

Can support higher level concepts like virtual memory

Easier to support threads

Cons. Fragmentation

Additional overhead

Recycling performance is based on free list management and the use of free block coalescing. Free block coalescing requires the ability to locate the blocks immediately preceding and following any arbitrary block. Some memory structures, such as boundary tag (see Donald Knuth's Fundamental Algorithms under References for more information on the boundary tag), allow this in O(1) time, while those that don't require an O(n) scan of the free block list. The current MADDEN memory manager uses boundary tags to allow O(1) access to previous/subsequent blocks and organizes the free list as an unbalanced binary tree. The result is allocation and recycling performance both in the $O(n \log n)$ to O(n) range.

Memory Overhead

emory overhead is the cost we have to pay to manage memory, since each allocation will cost some memory. Memory overhead is an important consideration, especially if the memory manager will be handling a large number of objects. The first decision is whether the memory map state should be stored internally or externally to the memory being managed. If it is stored internally, then the maximum number of allocated blocks does not need to be known in advance, but memory that is not directly CPU-addressable cannot be

managed. If it is stored externally, you must know the maximum allocated and free blocks in advance and set aside memory for this state, but address spaces that are not directly CPU-addressable can be managed.

MADDEN previously used external storage for the memory state, but this required additional overhead because it was impossible to predict accurately the maximum number of allocated and free blocks. We had to include a "safety factor," which turned directly into wasted memory. The new memory manager uses internal storage as shown in Figure 2, thus avoiding the entire issue. All allocations are 16-byte aligned and each block has 16-byte overhead. By limiting allocations to a minimum of 16 bytes, every block is guaranteed to have this much storage available. Therefore, when an allocated block is released, those 16 bytes can be used to organize the block into the free list.

It is worthwhile to digress slightly and consider the management of non-CPU-addressable memory. Because consoles are designed for low cost and high performance, they sometimes incorporate special-purpose memory that is not directly CPU-addressable. For example, the Gamecube has 16MB of auxiliary memory (ARAM) that supports only DMA access. A memory manager that stores its state internal to the memory it is managing cannot be used in such cases, while a memory manager that stores its state externally can.

While it may seem appealing to use external state storage in order to support all kinds of memory, our experience with MADDEN has shown this to be a mistake. Memory that is not directly CPUaddressable is normally used only for special types of objects, due to the complexity of using the memory, and often contains only a small number of objects. Therefore, MADDEN now uses a single, general-purpose internal storage memory manager for all CPU-addressable memory and an additional, customized external storage memory manager for any special-purpose case, such as the Gamecube's ARAM.

Fragmentation Control

emory fragmentation is a condition that occurs over time as memory is allocated and released and isolated free blocks form. This is not usually an immediate problem, but as more fragments form and they get smaller, the opportunity for an allocation failure due to fragmentation increases. Eventually, if fragmentation gets severe enough, an allocation may fail because no single free block is large enough to accommodate the request even though the total amount of free memory is (Figure 3). Therefore, a good memory manager needs to either take steps to limit the amount of fragmentation that occurs or be able to consolidate fragmentation periodically through garbage collection.

An allocation failure is often a lethal error for game code and must be avoided at all costs. While it is generally fairly easy to determine the maximum memory required by an application, fragmentation can make such a calculation meaningless. Allocation policy, free block coalescing and multiple heaps all play a part in minimizing fragmentation.

Free memory coalescing is a straightforward technique for limiting fragmentation that attempts to merge a newly released memory block with its neighbors. If the block immediately preceding or following the newly released block is also free, merging the blocks together results in a single, larger free block. This particular technique is almost mandatory, as without it, fragmentation occurs very quickly. In addition, this limits the size of the free list to the minimum number of blocks, which generally has a positive performance impact on allocation. Since this technique has no impact on how the memory manager is used externally, it is incorporated into most memory manager designs.

The choice of allocation algorithm has a definite impact on fragmentation. Unfortunately, because the size and timing of allocations and releases vary for every application and are often different from run to run of an application, it is impossible to say that a particular algorithm is always better or worse than another. However, in general, it has been found

GameDeveloper

ON THE FRONT LINE OF GAME INNOVATION

The *ultimate* resource for game development professionals

YES! I wish to receive/continue to receive a complimentary subscription to Game Developer.□ No					
X SIGNATURE (required)					
DATE					

	FREE SUBSCRIPTION	ON APPLICATION			
Γ					
REQUIRED	NAME				
REQU	TITLE				
ı	COMPANY (REQUIRED EVEN FOR HOME ADDRESSES)				
Ī	WHERE WOULD YOU LIKE YOUR SUBSCRIPTION SENT:	,			
	HOME ADDRESS	BUSINESS ADDRESS			
	CITY/STATE/ZIP	CITY/STATE/ZIP			
	PHONE/FAX	PHONE/FAX			
	EMAIL By providing your email address, you grant the Gama Network, a division of CMP Media, permission to contact you in the future via email concerning your Game Developer subscription and its affiliated properties				
	Please check the boxes below to let us know the types of information that may be of interest that we may send you via e-mail:				
	☐ Product and service offerings from other CMP Media prop☐ Product and service offerings from outside, relevant busin				
	THE CARD. INCOMPLETE FORMS C	ANNOT BE PROCESSED.			
in	ness?				
CE	RELATED INDUSTRY 11	18 ☐ Multimedia production 19 ☐ Corporate software development 20 ☐ Education/research			

PLEASE ANSWER ALL QUESTIONS, SIGN AND DAT

1. Which of the following best describes your company's primary line of bus (SELECT ONE ONLY)

GAME INDUSTRY

- 01 Independent 3rd-party game development
- 02 🗆 2nd-party game development with publisher
- 03 \(\sigma\) 1st-party game development/publishing
- 04 Game recruiter/agent
- 05
 Online game service provider
- CONTRACT GAME DEVELOPMENT SERVI
- 06 Animation/graphic arts
- 07 Programming 08 Music/sound
- 09 Script writing
- 10 Testing/QA

- 13
 Motion capture
- 14 🗆 Educational software development 15
 Tools/Middleware development
- 16 Hardware development 17 Usual simulations development
- 21 Analyst/venture capital/legal
- 22 Other (please specify)

2. Which of the following best describes your job title?

(SELECT ONE ONLY)

- VISUAL ARTS 01 Art Director
- 02 Lead Artist
- 03 Lead Animator
- 04 Animator
- 05 ☐ 3D Artist/Modeler
- 06 2D Artist/Texturer 07 Art Assistant

PROGRAMMING/ENGINEERING

- 08 Art Technician
- 09 🖵 Director of Development
- 10
 Technical Director
- 11 Lead Programmer
- 13 Al Programmer
- 12 Engine Programmer

- 14 🗆 Tools Programmer
- 15 Programmer
- 16 Network Programmer
- 17 Hardware Engineer

GAME DESIGN

- 18

 Creative Director
- 19 Lead Designer
- 20 Game Designer
- 21 Level Designer
- 22 Interface Designer 23 Writer

- 24 Director, SFX/Music
- 25 Composer/Musician

26 🗆 Sound Designer 27 SFX Engineer

PRODUCTION

- 28

 Executive Producer
- 29 Producer
- 30 Associate Producer 31 Project Lead/Manager
- 32 Uvideo Director
- 33 Asset Manager
- 34 \(\subseteq Editor
- 35 Localization
- 36 Licensing
- 37 ☐ QA/Tester
- 38 Documentation Development

BUSINESS/LEGAL 39 ☐ CEO/President

- 40 UP/Executive Manager 41 Marketing/PR/Sales
- 42 ☐ Analyst/Lawyer/Consultant 43 ☐ HR/Training Manager/Recruiter
- 44 🗆 IS/IT Manager
- 45 Uventure Capitalist/Investor

PUBLISHING

- 46
 Executive Producer/Producer
- 47 Content Acquisition
- 48 Product Manager
 - 49 ☐ Other (please specify)

3. What is your involvement in the purchase of tools and services for your job?

(SELECT ALL THAT APPLY)

- 01 \(\text{Identify need} \)
- 02 Evaluate brands or products
- 03 Recommend brands or products
- 04 Authorize or approve purchase
- 05 ☐ Not involved
- 4. Which products, services or technologies do you recommend, specify, or authorize for purchase?

(SELECT ALL THAT APPLY)

- **HARDWARE** 01 PC/Mac desktops
- 02 Workstations
- 03 \(\text{Monitors} \)
- 04 ☐ Servers/networking equipment 05 Consumer audio cards
- 06 Professional Audio/MIDI boards
- 07 Audio recording/editing equipment 08
 Consumer graphics accelerators
- 09 Professional graphics accelerators
- 10 ☐ Digitizing/3D scanning equipment 11 Video capture/playback boards
- 12 Video editing systems
- 13 Data storage/backup
- 14 ☐ DVD/CD-ROM burners
- 15 Game input devices (joysticks, etc.)
- 16 Productivity input devices (tables, etc.) 17 D Motion capture equipment
 - 18 Other Hardware (please specify)

- SOFTWARE
- 20 Programming environments/compilers
- 21 Cross-platform/porting tools
- 25 Testing & debugging tools
- tools
- 29 Game security 30 Image manipulation/photo enhance-
- 32 ☐ Motion capture software 33 ☐ Sound composition/editing

- 19

 Commercial game engines
- 22 Programming libraries/middleware 23 D rendering/modeling/animation
- 24 2D draw/paint tools
- 26 Programming utilities
- 27 Multimedia/authoring/prototyping
- 28 Project/asset management
- 31 Uvideo/film editing
- 34 Ursion control software 35 UVideo codecs 36 Audio codecs 37 Music libraries

- 38 🗆 Stock footage/clip media
- 39 3D model libraries
- 40 SFX libraries
- 41 🗆 Other Software (please specify)

SERVICES

- 42 Contract art/animation
- 43 Contract programming
- 44 Contract music/sound 45 Contract testing & QA 46 Disc replication
- 47 ☐ Post-production 48 Platform porting
- 49 Localization 50 🗆 Internet service providers
- 51
 Web hosting services 52 ☐ Internet/networking infrastructure

- 53

 Motion capture services 54 Dusiness (legal, marketing, accounting,
- PR, etc.) 55 ☐ Licensing representation 56 Professional representation (recruiters,
- agents) 57 Packaging production & design
- 58 Other Services (please specify)
 - **CONTINUE ON BACK**

5. Which products, services or technologies do you recommend, specify, or authorize for support/ optimization? (SELECT ALL THAT APPLY) 01

3D graphics accelerators 02 3D audio 03 Game peripherals 04 Online game services 05 Platform choice 06 ☐ Other (please specify)

6. On/for which of the following platforms are you developing your current/most (SELECT ALL THAT APPLY)

indows 95/98/ME indows CE indows NT/2000 nux ther UNIX ac OS	ON 01 □ 02 □ 03 □ 04 □ 05 □ 06 □	FOR 08 □ 09 □ 10 □ 11 □ 12 □ 13 □ 14 □	DVD Sega Dreamcast Nintendo 64 Nintendo Gamecube Sony Playstation Sony Playstation 2 Microsoft Xbox
Game Boy Color Game Boy Advance Palm OS Cell phone/WAP/Imode Other handheld OS CD-ROM		16	Online game channel (EA.com, etc MMPOG (UO, etc.) Web games Arcade/coin-op Interactive television Other (please specify) 07 □

7. What is the budget for your current game project? (SELECT ONE ONLY)

01 🗆	\$50 Million or more
02 🗆	\$20,000,000 - \$49,999,99
03 🗆	\$10,000,000 - \$19,999,99
04 🗆	\$5,000,000 - \$9,999,999
05 🗀	\$3,000,000 - \$4,999,999
06 🗆	\$2,000,000 - \$2,999,999
07 🗆	\$1,000,000 - \$1,999,999
08 🗆	\$500,000 - \$999,999
09 🗆	\$100,000 - \$499,999
10 🗆	Less than \$100,000

Publisher reserves the right to determine qualification for free subscriptions. Free subscriptions are limited to availability upon qualification, and are only offered to U.S. residents. If you are not employed in the interactive entertainment industry, or you are not a resident of the U.S., subscriptions are available for purchase by visiting our website at www.gdmag.com.

▲ FOLD COMPLETED CARD HERE, TAPE ▲ (DO NOT STAPLE) AND MAIL



22 🗆

23 🗆

24 🗆

25 🗆

26 🗆

27 🗆 28 🗆

29 🗆 30 🗆 31 🗆 32 🗆 33 🗆 34 🗆

> **PLACE** STAMP HERE

GameDeveloper

PO BOX 1274 SKOKIE IL 60076-8274



▲ FOLD COMPLETED CARD HERE, TAPE ▲ (DO NOT STAPLE) AND MAIL

Apply for your FREE SUBSCRIPTION today!

FOR FASTEST SERVICE FAX BOTH SIDES OF THIS FORM TO 1-847-647-5972



that "best fit" algorithms tend to exhibit the least fragmentation. See Johnstone and Wilson's "The Memory Fragmentation Problem: Solved?" in References for more information on the impact of allocation techniques on fragmentation.

A technique that requires more involvement from the memory manager user is to allocate blocks with similar lifetimes close to each other. If the lifetime of a memory block is known when it is being allocated, the allocation algorithm can attempt to place it by other blocks with a similar lifetime. When these blocks are released, they will be merged into a single large free block. The problem is that this requires the caller to specify both the lifetime of each block as well as the total size of the group of blocks of similar lifetime. As a result, a simpler version of this technique is usually implemented through the use of multiple heaps. By allocating blocks with similar lifetimes within their own heap, a similar effect is achieved

-12:	header pointer
-8:	type (head)
-4:	data length
0:	header data
+n:	data length
-12:	header pointer
-8:	type (used/free)
-4:	data length
0:	user data
+n:	data length
-12:	header pointer
-8:	type (used/free)
-4:	data length
0:	user data
+n:	data length
-12:	header pointer
-8:	type (used/free)
-4:	data length
0:	user data
+n:	data length
-12:	header pointer
-8:	type (tail)
-4:	data length (0)

FIGURE 2. Memory structure image.

though there are generally practical limitations on the number of heaps that can be effectively utilized.

Debugging Support

ne of the main benefits of writing a memory manager is the ability to add capabilities that are not provided in the supplied memory manager. By adding some debugging features you can make sure that you are managing the memory rather than stepping in the heap.

To avoid stepping in the heap, you'll need to be able to see the heap. Adding information and tools to help visualize memory usage can be a big help when managing memory. This can be done in a number of ways.

The debugging techniques used by the MADDEN memory manager include the ability to audit memory usage, collect usage statistics, check for consistency, and follow the memory map from a debugger. Auditing is really nothing more than being able to mark a group of blocks when they are allocated and later check to see if they were all released. Usage statistics, such as maximum number of allocated and free blocks, as well as maximum allocated memory, are valuable for evaluating the memory utilization of the application.

Consistency checking goes through the memory map and makes sure there are no detectable errors in it. This means verifying block sizes, checking boundary tags, ensuring matching block counts, and so on. By performing consistency checking

256 used bytes	
32 free bytes	
512 used bytes	
64 free bytes	
96 used byte	
80 used bytes	
32 free bytes	

FIGURE 3. Although 128 free bytes are available, the largest possible allocation is only 64 bytes due to fragmentation.

prior to every allocation and release (in a debug build), certain forms of memory corruption can be detected quickly.

Also, the memory map contains ASCII debugging information that can be viewed from within the debugger. Prefixing every allocated block with a text string indicating the module and line number of the caller that allocated the memory greatly assists when debugging after a crash.

Naming the memory blocks is a necessity. Imagine going to a store where every item was in a plain box that only had its UPC code and box size printed on it. Finding the item you wanted to buy would be a big challenge. Likewise, if you have a list of memory blocks that only contains their address and size, locating memory blocks is going to be difficult unless you give those memory blocks names. Using the source file name and line number where the allocation occurred would be a good start. In C, this can be accomplished quite easily through the use of macros and the C preprocessor.

Now you can recognize your blocks in a crowd, but it sure would be nice to know who they hang out with. By adding a group label or flags, you can group your allocations at the conceptual level rather than being limited to grouping your blocks by source file. This way you can know that a memory block that was allocated is really being used by your main character, even though the actual memory allocation occurred in your texture library.

Figure 4 shows an example memory dump as seen by a (slightly drunk) debugger. Including debugging information in textual form within the memory map allows you can make sense of it from the debugger. For example, if you had an invalid address access at 0x007cfea4, you could find that address in the debugger and page around it to see that it occurred in the Joystick Driver library and that the memory in question was allocated by pJoyDrvr.c at line 225.

With all this information available, you will need to find ways to view that information that can help you when managing your game. A linear list of memory blocks can be useful for spotting

potential memory fragmentation, while a list of memory blocks sorted by name can be useful when you have memory leaks. If you do find a memory block that has leaked, you will know from its name exactly where the allocation occurred. With group labels you can print out compact memory maps that show how each conceptual module in your code is using resources. By tracking this throughout the project, you can easily spot modules that are using more or less memory than you had budgeted in your original design.

You can also create functions to check whether groups of memory allocations have been freed. This can help prevent memory leaks if you know that in certain situations some groups have no allocations.

Keeping track of worst-case situations is also important. Have the memory manager save the lowest amount of free memory it has ever encountered (update this every time an allocation occurs). If you are using some sort of node system to keep track of your memory blocks, keep track of the highest number of nodes that the memory manager has ever used so that you know if you are close to running out of memory blocks.

Sentinels added to the beginning and end of memory allocations can help detect overflow situations that would normally corrupt other data in memory. If the sentinels don't check correctly when the memory is freed, then some code has been bad.

Filling memory with recognizable patterns can be extremely useful. We use a different pattern for unallocated, allocated, and freed memory. This way, we can tell in the debugger at a glance what the situation of a particular piece of memory is. When someone forgets to initialize memory they allocated properly, the "allocated" pattern shows up.

You can also have functions that scan free/unallocated memory and make sure that it all still matches the prespecified patterns. If it doesn't match, some code out there is incorrectly writing to memory locations that it doesn't own.

Finally, make sure that you set up an extra heap for debug memory and put all this extra debug information there. You want your game memory to be as similar as possible between a debug and final build.

Platform Specifics

memory manager presents a logical view of memory in the same way that a file system provides a logical view of a disk drive. Most often, memory managers are concerned with managing dynamic RAM of some sort. Some console makers like to make things more interesting by providing a relatively large main memory but also scattering other smaller chunks of RAM around the system. The memory manager allows us to abstract away the physical details of the memory subsystem and deal instead with a

nice, logical model of memory. For example, on the PS2 we don't necessarily need to know that the first megabyte of RAM is reserved for the operating system. It's enough that the memory manager knows. By abstracting away some of the details of the physical memory system, our game can become more platform independent.

Most console hardware has alignment requirements that, not so surprisingly, differ from platform to platform. Many platforms require anywhere from 4- to 64-byte alignment for buffers in graphics rendering or file IO. Each type of hardware might need the memory manager to be tweaked to better fit the needs and abilities of the platform. Often this information can be passed to the memory manager using a heap setup structure.

Finally, you should be wary of thirdparty libraries that may use malloc and free, effectively bypassing your memory manager's interface. The printf function in the PS2 standard C library uses malloc and free; our solution was to write our own printf function.

Hardware Support

n most modern computers, the issue of fragmentation has been greatly reduced by the use of a hardware-based paged memory manager unit (PMMU). Obviously, the fact that virtual memory provides an application with lots of addressable memory means that even an inefficient memory manager can be used. However, the more interesting point is that the PMMU without any secondary storage can dramatically help with fragmentation. The PMMU takes a very large address space (larger than the physical RAM it represents) and maps it onto physical memory. Obviously, this mapping is not one-to-one, but rather it maps a subset of the memory space onto a "working set" of memory pages.

The key impact of using a PMMU in terms of fragmentation is that when a memory block is released, any portion of that block completely spanning one or more PMMU pages can be remapped by the PMMU. The result is actually two forms of fragmentation: address-space fragmentation and memory fragmentation. While this effect might seem to make a bad problem worse, it actually simplifies things. Because the PMMU provides a large address space, the address-space fragmentation can be largely ignored. Instead, the allocation algorithm concentrates on minimizing

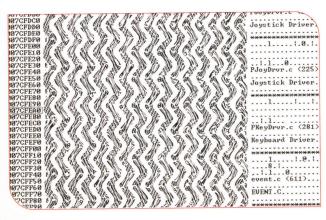


FIGURE 4. Memory blocks with debug information as viewed in the debugger.



memory fragmentation by first attempting to utilize the unused portions of allocated memory pages (pages that contain some allocation but are not completely full) before remapping unused memory pages back into memory.

Managing Your Memory Manager

emory managers are like most reallife managers. You have to keep them under control, or else they tend to wander off without adequate information and make strange decisions. Your memory manager sometimes needs your help to stay on the right track too. Let's look at some common problems that can occur and how to work around them.

Fragged again. One of the major side effects of hand grenades and memory managers is fragmentation. Previously, we discussed fragmentation in the context of designing the memory manager to avoid or reduce the effects of fragmentation. However, there are also some application-level techniques that can reduce fragmentation.

The use of static allocation (memory defined within a module with storage allocated within the application image) avoids the memory manager completely and thus avoids fragmentation. Of course, this usage really only makes sense when a single object will be represented and when the lifetime of that object is approximately the duration of the entire application. In such cases, static allocation can provide a benefit by limiting utilization of the dynamic memory manager to those memory objects that are truly dynamic.

Another strategy that relies entirely on the memory manager user is to perform submanagement of memory based on specific program knowledge. For example, if a module knows that it needs to allocate X objects of size Y, it may be far more efficient for the user to allocate a single block of X * Y bytes and perform its own management within that larger block. By reducing the number of blocks that the memory manager has to deal with, the user has generally made the job of the memory manager easier. There is,

of course, a caveat. Depending on the amount of fragmentation and the size of X * Y, it is possible that the application could find itself in the situation where an allocation of X * Y fails due to fragmentation, whereas X allocations of Y would have succeeded. We also try to discourage this practice when possible, as there is code-maintenance overhead.

One way to help avoid memory fragmentation is to always free memory in the opposite order from which it was allocated. In fact, if you were always able to allocate and release in such an order, you would never have memory fragmentation. Realistically, it's not possible to follow this practice all the time, but doing it as much as possible will help.

Memory fragmentation is going to occur, and at some point you will probably run into a situation where it causes a problem in your game. You might have fragmentation that is occurring in one part of your game that is causing a memory allocation to fail in a totally unrelated area. Fragmentation might even manifest itself as a situation where your game will fail only when arriving at part of your game through a specific path (that causes the fragmentation). If you don't have some advanced form of garbage collection, you are going to have to use other, more crude methods to limit this problem.

One possibility is to change the code that is causing the fragmentation to use a different allocator so that it doesn't cause fragmentation. A common way to accomplish this is to have an allocator that allocates from the other end of memory. Depending on your game, fragmentation can become more problematic over time (especially if your allocations don't have close locality of lifespan). You can use brute force to minimize these effects, such as shutting down and restarting code modules between levels as a brute-force garbage collection technique.

Release builds. When running a release build, there isn't any debugging information in the game, as it will consume extra memory. But you still need a way to know where the game runs out of memory. For MADDEN, we assume in the code that we will never run out of

memory. If the game does run out, it will crash and optionally dump some memory info. With a special build, we display some partial information about the memory manager and the game state so that we can determine if it ran out of memory because of fragmentation or other reasons.

Getting on the Field

hen talking about memory management, programmers often resort to words more often associated with cow pastures than games. Terms like heaps, stacks, and chunks are thrown around like flies buzzing around you-know-what. To see how important a memory manager is to a game, you have to get past the abstract poo. A good memory manager allows you to have more animations, more characters, more textures, more sounds — in short, more of everything that your game-playing customers love.

In this article we have described some of the issues that may come up in writing and using your own memory manager. After years of writing and rewriting memory managers for the MADDEN series, one piece of advice that bears highlighting is simply to make sure you schedule adequate design time on this very important piece of your system; you'll be glad you did.

REFERENCES

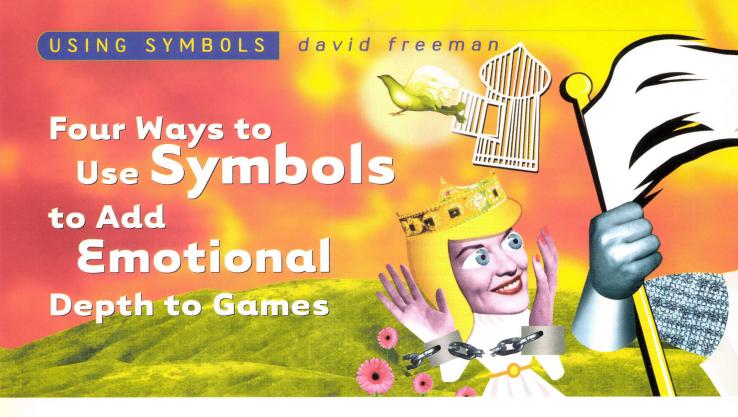
The Memory Management Reference Beginner's Guide:

www.memorymanagement.org/ articles/begin.html

Johnstone, Mark S., and Paul R. Wilson.
"The Memory Fragmentation Problem:
Solved?" In Proceedings of the
International Symposium on Memory
Management. ACM Press, 1998. pp.
26–36. www.cs.utexas.edu/users/
wilson/papers/fragsolved.pdf

Knuth, Donald E. *The Art of Computer Programming*, vol. 1, *Fundamental Algorithms*. Reading, Mass.: Addison-Wesley, 1973. pp. 435–444.

www.gdmag.com 35



AX PAYNE. ELITE
FORCE. THIEF. ICO.
DEUX EX. ODDWORLD.
MEDAL OF HONOR.
BALDUR'S GATE. The
more recent Final Fantasy games. More
and more developers are pushing the
game design envelope, forging new entertainment experiences and art forms that
draw on the roots of traditional gaming,
but also partake of more sophisticated
storytelling and characterization.

As the production values in games continue to soar, the trend toward equivalent advancement in storytelling is inevitable. For game designers involved in creating each successive advancement, these are exciting times.

Remember in *Braveheart* when Mel Gibson charged into battle holding a handkerchief his wife gave him before she was murdered? That handkerchief is a symbol. This article will explore four different ways to use symbols to evoke emotional response from an audience.

But first, let's look at some of the fundamental issues relating to the role of emotion in games.

Why Put Emotion into Game Stories?

This is an important discussion, and probably one that deserves its own article. But, in a nutshell, other than the inherent joys of creating a rich work of art, the reasons also boil down to potential profits.

First of all, many more people watch film and television than play games. Most will never be lured into playing games until games begin to offer the emotional range and depth of the entertainment that they're used to enjoying. Also, a more involving game experience means better word of mouth and more buzz. The press likes to write about these kinds of games, which results in more sales. Seeking out better profits also means staying ahead of the competition. Certain game developers

are working hard to advance emotion in gaming. Those creating games with stories and characters without investing in putting emotional depth into their games will find themselves further and further behind, and their games will be eclipsed. And, the better game visuals get and the more games look like films, the more people will want to compare them to films. Thus, weak writing and shallow emotional experiences in games featuring stories and characters will increasingly stand out negatively in consumers' minds.

Many of the challenges that designers face in creating emotionally rich game experiences have already been addressed in other media. Traditional screenwriters, deprived of the game designer's ability to actually insert an audience into a film, have figured out perhaps thousands of ways to induce emotional involvement. Game designers will want to test the applicability of these techniques to their new games and modify them so they'll work within an interactive experience.

A big part of successful communication between a writer and his or her audience is writing outside of the audience's conscious awareness. No one expects the game player to pick out every sound used in a game's sound design, nor every instrument utilized in a piece of music, nor every tiny shadow. So too, an extraordinary amount of what a writer

DAVID FREEMAN | David contributed to the script for The MATRIX sequel game (in production at Shiny Entertainment). As a writer and producer, David has had scripts and ideas bought or optioned by MGM, Paramount, Columbia Pictures, Castle Rock, and many other film and television companies. David teaches "Beyond Structure" (www.beyondstructure.com), a popular Los Angeles—based screenwriting class, which has been taken by writers from many top films and television shows, as well as by many well-known game designers. He welcomes your thoughts on this article at freeman@dfreeman.com.



Illustration by Ben Fishman

does is designed to affect a game player emotionally but not be consciously noticed. This article will focus on the use of symbols, which are almost always employed in a way so that they're just on the edge, or preferably just outside, of a game player's conscious awareness. A workable rule of thumb is that no more than 25 percent of the players who come upon a symbol should be consciously aware that it actually is a symbol.

The five arenas of "deepening." I use the phrase "deepening techniques" to describe all those writing techniques that impart a sense of depth to a piece of dialogue, a character, a relationship between two or more characters, a scene, or a plot. Other words that mean something similar to deepening include poignancy, soulfulness, layers, and emotional or psychological complexity. When people talk about these things, they're talking about what I call emotional deepening. Symbols are always a deepening tool.

One game designer who has taken some of my story and writing workshops pointed out that to focus on more subtle or sophisticated techniques such as the use of symbols is putting the cart before the horse. Many game designers might benefit from learning more basic techniques for creating rich, complex, and compelling characters and natural dialogue. This is true. But one nice thing about symbols is

that, with very little effort, you can easily and precisely enhance the depth of your scenes and plots.

When you create a symbol, you're not trying to create an intellectual puzzle in which the player tries to figure out what the symbol means. Such an intellectual exercise would work directly against the goal of increasing emotional immersion. Instead, symbols, when employed artfully, should evoke emotions — even though, when you do your work well, most players won't consciously notice the symbols that you use. It's not necessary for a game player to notice a symbol in order to be emotionally affected by it.

It's certainly O.K. that a small percentage of players who consciously notice your symbol might stop and think about the symbol's meaning or meanings. But it's only acceptable if, at the same time, the symbol generates in those players an emotional experience as well. Following the guidelines in this article will help ensure that this is what the player actually experiences.

Another advantage to using symbols in game design is that games often offer an opportunity that films do not. In film, symbols, when used artfully, enhance emotional depth. As we'll see, when used in games, symbols can not only perform this function, but can also be used or given a function in gameplay as well.

Symbol Type #1: Symbol of a Character's Condition or Change in Condition

his use of symbols is what I call a scene-deepening technique, because you use it in a specific scene and might never use the same symbol again. Its use can be either visual or verbal, meaning that there must be either something visual on screen or something said by one of the characters that reflects what an on-screen character is going through emotionally.

episode of Star Trek: Voyager, Captain Janeway (Kate Mulgrew) finds herself in an extended battle with the captain of a rogue Federation ship. The captain and crew of that ship are killing harmless aliens in order to use the chemicals in the aliens' bodies to propel their ship. But Janeway herself becomes so obsessed with stopping the rogue captain at any cost that she crosses the bounds of ethics and good judgment and imperils her crew. This conflict generates a series of arguments with Chakotay (Robert Beltran), her first officer.

A metal plaque with the words "U.S.S. Voyager" falls off of Voyager's bulkhead during the battle with the rogue ship. This plaque is a symbol that

the spiritual core of Voyager — including the moral codes of the Federation, the Starfleet tradition of honor and humanity, and the moral center of the people who uphold these codes and traditions — has been damaged. It's a symbol of Janeway's and Chakotay's conditions or changes in condition.

The plaque falling off of the bulkhead affects us emotionally. If viewers make only an intellectual connection between the plaque and the abandoned Federation values, then the writer hasn't been artful enough in his or her creation of the symbol.

Example *2: Visual. The 1957 war film Bridge on the River Kwai won many Academy Awards and still stands up as a masterpiece. Alec Guinness plays Colonel Nicholson, who commands a group of British soldiers captured by the Japanese and forced to work as slaves in a POW camp in Burma. I won't reiterate the convoluted plot, but in short, due to his ego, Nicholson has his men help the Japanese build a strong and beautiful bridge. In effect, he has helped the enemy. But, near the end of the film, during a battle at the bridge, he has a powerful revelation, and says, "What have I done?"

At that exact moment, he reaches up and touches his commander's cap. This is a symbol of the character's condition or change of condition. His touching the cap is a symbol of his changing back to becoming what he once was — an honorable British soldier.

An explosion goes off nearby that knocks him to the ground, wounded by shrapnel. When he stands up, his cap lies on the ground, but he's too dazed to notice immediately. He reaches for the top of his head and realizes that the cap is gone. He then bends down and picks it up off the ground. His reaching toward his head for the cap, and then his picking it up off the ground, again is the same kind of symbol, signifying that he's become the honorable man he once was.

He puts his conversion into immediate action. As he dies from the shrapnel wound, he directs his fall onto a dynamite detonator, which in turn blows up the bridge he had so painstakingly built.



In Sony's Ico, the main character's quest to save a beautiful girl with mystical powers contains symbols that engage the player's emotions and affect gameplay.

As was the case with the *Voyager* example, most people in the audience wouldn't consciously notice this element. And yet it would still contribute to the depth of the audience's emotional experience. It's a strange moment for a writer when he or she realizes that a great deal of writing involves trying to create emotional effects that no one will consciously perceive, perhaps ever.

Example #3: Verbal. Perhaps you saw the provocative film *American Beauty*, in which Wes Bentley plays Ricky Fitts, a teen without fear of social pressures, who has an honest appreciation for the beauty all around him. He seems, in some ways, to be enlightened.

Contradicting his supposed enlightenment is the fact that he sells drugs, is completely emotionally detached, and is fascinated by death. In fact, his veneer of serenity is what I call a "mask," or a false front. (Masks, in all their various forms, are very sophisticated character-deepening techniques.)

At a certain point in the film, Lester Burnham (Kevin Spacey) drops in on Ricky to buy some dope — in particular the really potent stuff that he'd smoked

with Ricky a few nights earlier. Ricky pulls out a bag of dope and explains that it's "... top of the line. It's called G-13. Genetically engineered by the U.S. government. Extremely potent. But a completely mellow high, no paranoia."

LESTER: "Is this what we smoked last night?"

RICKY: "This is all I ever smoke."

Why is this a verbal symbol of a character's condition or change of condition? Because Ricky, unknowingly, has just described himself. Ricky had been a passionate young man, until his father, as punishment, had him committed to a mental institution for two years, where he was heavily drugged. This experience broke his spirit. So Ricky himself has been government-engineered, and his fake serenity (his mask) is that of a "completely mellow high." But like all chemical highs, the effects aren't real.

Example #4: Verbal. Sometimes, in the television business, you need to write a sample script just to show that you can adapt your writing style to different shows. I recently wrote a sample *X-Files* script. In the story, Mulder no longer fits in professionally with Scully and

Doggett. He had always been driven in his paranormal quests by the search for the truth about his missing sister. But, with that case solved last season, he no longer has a dream or ambition to push him forward.

In the middle of a conversation with Scully, Doggett, and Skinner, Mulder notices Skinner's office clock. Checking it against his own watch, he says, "Is that clock right?"

No one responds to the question — the conversation merely proceeds. (Quite frequently, in dialogue, not every statement or question gets a response.) Why the throwaway line about the clock? It's a symbol of Mulder's condition or change in condition. In this case, it symbolizes that he's out of sync, or out of step, with all the others. In effect, his time has passed.

Will anyone reading the script consciously note that line of dialogue? Unlikely, any more than they would note Wes Bentley's line in *American Beauty* about the government-engineered pot. As with the other examples, the symbol operates outside of the audience's conscious awareness.

Game example. In the game ICO, a boy in a faraway land helps lead a beautiful girl with mystical powers out of a towering castle where both are trapped. He bravely overcomes many terrifying obstacles in his journey, which is more focused on freeing the girl than himself.

Near the very end, he gets a magical sword that crackles with a kind of spiritual electricity. This is a symbol of the boy's condition or change in condition. It symbolizes that he's attained a level of power; the demonic creatures that once attacked him now flee him and the sword. And it symbolizes that he now belongs with the girl, for the electricity that the sword exudes looks exactly like the mystical energy that the girl can wield when she needs to, and which has the same magical abilities.

Since the boy uses the sword to accomplish his final tasks, this is what I call a usable symbol. It serves double duty by both working to deepen the emotional experience and also playing a role in gameplay.

Hypothetical game example *1. Let's say we have a sword-and-sorcery game in which, during a fight to save some villagers, the wisest and most beloved village elder is killed. The villagers are stunned. A cloud could pass in front of the sun at that point, throwing a shadow over the village (during either a cinematic sequence or gameplay). The shadow would symbolize the villagers' sadness — and perhaps yours as well, if you had found the old man endearing (and you would have, if the character was rich enough and the dialogue was compelling).

Hypothetical game example #2. After great effort and many struggles and battles, you have attained the highest rank a warrior can attain. At that moment, an eagle flies diagonally overhead in the sky. It's a symbol of your lofty achievement.

It's important to reiterate here that it doesn't matter if no one consciously notices the impact of these symbols. They deepen the experience nonetheless.

Symbol Type #2: Symbolic Subplot

sually at least one of the characters (although sometimes more) in a story has what I call an emotional fear, limitation, block, or wound. Quite often, this person is the lead character, although not necessarily.

In the first Star Wars movie, Luke Skywalker had to learn who he was (a Jedi knight), Han Solo had to learn responsibility and how to act as a member of a group (instead of operating solo), Princess Leia had to learn to be vulnerable in love, Obi-Wan had to learn he could still make a difference, and C-3PO had to learn courage. Each of these characters was forced to confront their respective fears, limitations, blocks, and wounds (FLBWs, for short).

Usually, the character doesn't know he or she has an

FLBW. If you pointed it out, the character would probably disagree; in fact, they're usually quite oblivious. It's unlikely, for instance, that Han would have agreed with you if, at the start of the film, you accused him of being unable to function as part of a team. It's unlikely Luke would have agreed if, at the start of the film, you accused him of having no idea who he was.

A character's path of growth through his or her FLBW is a rocky one; quite often the character resists growing. A character's path of growth through the FLBW is called a character arc. In many stories, some of the most compelling emotional moments are wrapped around a character's process of wrestling with and eventually growing through his or her emotional fear, limitation, block, or wound.

Some writers insert a symbol into the story that represents the character's arc. That is, as the character changes and grows, the symbol changes right along with the character. Therefore, a symbolic subplot is a plot-deepening technique because it continues throughout all or most of the plot (unlike the symbol of the character's condition or change in

condition, which occurs in a sin-



Example #1. In the new *Star Trek* series, *Enterprise*, one of the crew, Ensign Hoshi Sato (Linda Park) is a woman with extraordinary linguistic abilities. In one of the early episodes, she's having a hard time adapting to life on a starship. She wants to go home, back to Earth.

She has brought a pet along with her — a yellow slug. The slug isn't doing well aboard the ship. Environmental conditions threaten its health.

By the end of the episode, after discovering how much the crew needs her, she has made her peace with being in space. She drops the slug off on an Earth-like planet, where it will survive just fine.

Thus the slug is a symbolic subplot. The slug not doing well in space equates with Sato not doing well in space. The slug being put on a new planet and doing well there thus equates with Sato surviving and thriving away from Earth.

With a symbolic subplot, the audience can stay abreast of a character's progress in his or her character arc just by checking up on what's happening with the symbol. Just as in the case with the symbol of a character's condition or change in condition, a symbolic subplot may or may not be consciously noticed by the audience or game player.

Let's revisit the example from the *Enterprise* episode. In this case, unlike most, we are quite aware that the slug is a symbolic subplot, for the doctor on board the ship even points this out to Ensign Sato. While speaking to her, he compares her difficulties to those experienced by the slug.

This bit of dialogue violates the guideline of having the symbolic subplot operate just outside of most people's conscious awareness. In my opinion, this was a mistake. The slug symbol would have generated more emotion if it hadn't been pointed out to the audience. "Look, here's a symbol" is usually not the best way to go. However, as every writer knows, to every guideline there are always successful exceptions.

Example #2. In the film *Wonder Boys*, Michael Douglas plays a character who

wrote a great novel decades ago and is now a washed-up creative writing professor at a prestigious liberal arts college. His life's a mess. He's depressed, and he's been working forever on a sprawling novel that he hasn't shown to anybody.

The symbolic subplot is the novel he's writing. The novel is analogous to his life. We learn that the he's been working on the book for decades. Then we learn that it's a sprawling jumble, with plotlines going off in all directions but no focus, just like his life. It comprises tons of details without a unifying thread, just like his life.

Further along in the film, the pages of his manuscript — the only copy he has — are blown to the wind (symbolic of his life falling apart). Later still, when someone asks him what the novel was about, he can't answer — meaning he has no idea what his life is about. By the end, once he feels his life has again assumed meaning and direction, he starts a new novel, a novel that has power and focus.

Using this technique in games. Trying to build in a character arc for your player opens up a can of worms, because in a symbolic subplot, the changes in the symbol reflect the changes that your character undergoes as he or she progresses through the rocky path of his or her character arc. And how do you manage how a character goes through a character arc when that character is controlled by the game player?

This question takes us right to the cutting edge of story-based games. To explore all the ways in which game designers are tackling or could tackle this problem would be an article in itself, if

not several. Furthermore, it opens up another problem. On one hand, how do you tempt players into seeing themselves in a role and making decisions appropriate to that role? On the other hand, how do you allow players to play the game the way they want to play?

Still, this is one direction in which story-based games are moving. For instance, let's take Raven Software's action-adventure game STAR TREK: VOYAGER — ELITE FORCE. The game tries to create a character arc for Alex, the main player character (what I call a "first-person character arc," since the person who's supposed to undergo emotional change is the player). The attempt to cast the player as the Alex character, thereby helping the player to experience character growth during the story, is done through a variety of methods: observation of a character's behavior and speech during cinematics; watching how other characters respond to the player character; hearing the words coming out of the player character's mouth (what I call "self auto-talk"), spoken in Alex's voice and with his personality; and the player's changing responsibilities as the game progresses.

While these first-person character arcs are a fascinating and critical area of discussion, I'll bring the subject back to where we began. How can a designer use a symbolic subplot to deepen a plot by echoing a player's first-person character arc?

Let's imagine a game in which the player is a samurai swordsman. He's a master of many weapons. Armed with a full range of finely honed steel instruments of death, he leaves his samurai master's training to rescue his master's niece from an evil warlord. This mission will set a much bigger plot in motion.

The obvious character arc follows the player character from his origin as a novice swordsman to becoming a master himself. Because this is the most typical character arc, let's toss it out. As I often tell my writing students, when it comes to characters, lines of dialogue, scenes, or

plots, a good general guideline is, "Find the cliché, then throw it away." (This guideline also dictates that the master not be a clichéd wise Asian character either.)

So let's make our character's arc to "attain a spiritual connection to the universe." As the samurai character attains spiritual wisdom or abilities, perhaps the world will start looking different in some way. Perhaps he'll be able to perform extraordinary moves akin to those demonstrated by the fighters in *Crouching Tiger*, *Hidden Dragon*. Could you give this character arc a symbolic subplot?

Hypothetical game example #1. Perhaps the samurai's master has given him a sword. It makes a harsh, ringing noise when swung. But as the player character progresses along the character arc, the noise becomes beautiful and harmonic.

Hypothetical game example #2. Suppose the player character recharges his life force by returning to a beautiful little bamboo meditation hut suspended over a small stream. In the beginning of the game, the stream is muddy. But as the player progresses along the character arc, the stream gets clearer and clearer.

In either of these two examples, the player may or may not notice the change in the symbol. This situation is just what a game designer generally wants: a symbolic subplot that works just at the edge of the player's conscious awareness or just outside of it.

Making usable symbols in gameplay. In the first example, perhaps when the sword makes its most beautiful, harmonic sound, something extraordinary happens. A frail old man in the village is, in fact, much more than the peasant he appears to be. When he hears that beautiful sound, he knows the samurai is spiritually ready and gives the player character some special weapon, amulet, potion, or secret that is essential to the accomplishment of the game's final and most dangerous task. Or, taking a cue from Ico, perhaps it's only when the sword makes this beautiful sound that it's fully charged and thus useful against the final and most formidable enemy.

You could also find a way to turn the river (in the second example) into a usable symbol. Maybe the master built the meditation hut over the river and

imbued it with magic of which the player character is unaware. Let's say the master dies during the course of the game. But, when the character arc is complete and the stream becomes clear, the master's face can be seen in the river, from which he dispenses advice that is crucial to accomplishing the game's final tasks.

A symbol doesn't need to be used in gameplay to justify its being there, for its main purpose is to enhance the depth of the emotional experience. However, a symbol that can also function as an element of gameplay obviously represents an opportune situation.

Game example. In the game AIDYN CHRONICLES: THE FIRST MAGE, one of the player character's close friends is an NPC who's a reluctant knight. Though the knight has sworn off the violence of battle, he's continually forced to fight for his king, for honor, and to support an honorable cause. He carries a pole bearing the banner of the kingdom he serves. As a tool of gameplay, the banner has certain protective functions. Because of this, the banner is often ripped in battle, symbolizing that the knight's heart is torn every time he violates his decision to abstain from fighting. Furthermore, the banner, when torn, prompts discussions by the knight and those around him as to the ethics of his fighting in battle versus being a man of peace. The banner is a symbolic subplot, indicating, at any given moment, the knight's state of mind as he wrestles with the decision to be, or not to be, a warrior.

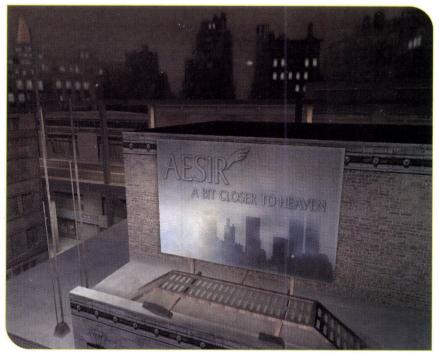
This is one of those examples in which a symbol serves a double duty. Not only does it deepen the emotional experience, but it also is a usable symbol with a function in gameplay.

Symbol Type #3: Foreshadowing

oreshadowing is another plot-deepening technique. Although it only appears in one specific scene, it prepares us for a later plot development. In foreshadowing, once again you're creating a



In the first-person action game STAR TREK: VOYAGER — ELITE FORCE, players assume the role of a character named Alex. One way we know Alex is growing in maturity and wisdom is that he's given more and more responsibility for the Elite Force team.



This billboard is littered across the rooftops in MAX PAYNE. As the story evolves, so does the significance of the associations the slogan carries, heightening the player's emotional involvement in the game.

symbol that usually operates outside the conscious awareness of the player or audience. The symbol, or what occurs to the symbol, suggests something that will occur later in the story to one of the main characters — usually something bad.

Example. In the film *The Shawshank Redemption*, Tim Robbins plays a man who has been unjustly sent to prison. There he runs afoul of the warden, and the two become enemies. Later in the film, another man who has information that could clear Robbins is sent to the prison. The warden finds out about this and asks the man to step out with him into the prison yard at night. The warden grills the new prisoner, who confirms his knowledge of information that could help Robbins.

The warden, finished with his inquiry, tosses his cigarette on the ground and steps on it to put it out. He walks away, and the prisoner is shot from an unseen source in a guard tower. The extinguishing of the cigarette was the foreshadow-

ing that the prisoner, or at least the information he had, was going to be snuffed out. As such, it evokes an ominous feeling when we see it happen.

Hypothetical game example. Let's go back to our samurai swordsman. His master has a bonsai tree that is 150 years old, cultivated and handed down to him by his own master, who is long since deceased. The samurai's master has used the careful cultivation of the small tree to perfect his patience.

Then, either during a cinematic or during gameplay, the villain destroys the tree. This would foreshadow the master's impending demise.

The bonsai tree could also be turned into a usable symbol with a function in gameplay if its magic heals the samurai when he's injured or restores his life force when it's been depleted. Thus, the tree's destruction would not only foreshadow the master's death, it would also affect gameplay by depriving the samurai of a source of healing and thus increasing his jeopardy.

Symbol Type #4: ASymbol That Takes on More and More Emotional Associations

his is another plot-deepening technique, as it too tends to extend throughout an entire plot. It can be either a visual object or a verbal phrase.

One symbol of this type is a very familiar one: the American flag. What does the flag mean? It means a lot of things: democracy; courage; the right to live the life you choose; freedom of speech and religion; a nation ruled by law; Yankee ingenuity; and more. Yet when we look at the flag, we don't consciously think of all these things, we just experience the emotions that these associations evoke in us.

When a symbol reappears over and over again during emotionally charged moments, some of the emotion rubs off on the symbol, and the symbol thus takes on more and more emotional associations as the plot advances.

Visual example. In the film *Braveheart*, Mel Gibson plays William Wallace, a historical revolutionary leader in Scotland. There's an interesting symbol used throughout the film — a thistle, and a handkerchief with a picture of a thistle sewn into it. This symbol takes on more and more emotional associations as the film goes along.

When Wallace is young, a little girl, Murron, gives him a thistle at the funeral of his father and brother, who have been killed by the English. So the thistle is associated with love. When they're older, the two begin dating, and he gives her back this same, dried thistle. Once again it is associated with love. When Murron marries him, she gives him a handkerchief with a picture of a thistle embroidered on it. It is still associated with love.

Later, Murron is murdered. Had this been the only way the handkerchief had been used, whenever Wallace looks at it with sadness, we would understand and feel his personal anguish. It would evoke in him (and in us) emotional memories and feelings about her uniqueness, the beauty of their love, and the sadness of her passing.

At this point, we could call this a highly personal symbol, as it would be highly personal to him for reasons we can understand and which move us too. A highly personal symbol, and a character's reaction to it, can be an effective way to evoke a lot of emotion. It's a character-deepening technique. However, in *Braveheart*, the handkerchief goes on to take on more and more emotional associations throughout the plot, and so it becomes a plot-deepening technique.

After killing the English magistrate who had murdered Murron, Wallace stares at the handkerchief. By now it's begun to be associated with revenge. The handkerchief will be with him as he becomes a leader of the Scots in their fight for independence, so it eventually comes to be associated with freedom. And finally, after Wallace is killed, wishywashy landowner Robert the Bruce takes up the fight. Robert leads his men into battle holding the handkerchief, which is now associated with courage.

Throughout the film, the handkerchief with the thistle keeps reappearing, always during emotionally charged moments and always associated with love, revenge, freedom, or courage. By the end, the handkerchief is simply saturated with emotional associations, sort of like the American flag. An important point to make here is that when we see the handkerchief in *Braveheart*, we don't consciously think about all of these meanings and associations. Instead, the handkerchief evokes feelings in us from the many emotional experiences with which it has come to be associated.

Hypothetical game example: Visual. Let's say you're designing a game with a Tolkien-like story. (Yes, it's overdone, but we're just talking hypothetically.) So you've got your meek, Hobbit-type player character going up against a fearsome enemy with supernatural powers. Maybe the player character's motivation is that the villain wiped out his family. His father had given him a pendant with his family crest, handed down through the generations.

The first time we see the pendant is in a cinematic, when the father, as he lies



dying, gives it to the son. So the pendant is associated with love. As the player character goes on his quest to bring down the villain, he can recharge his life force (if he doesn't do it too much) by clenching the pendant. So the pendant comes also to be associated with life. At some point the player character needs to give the pendant to a fallen, dying friend, to save her by recharging her life force. Now the pendant is associated with the act of self-sacrifice for a friend. And if the pendant eventually comes back to the player character and gives him a decisive superboost of life force for the final battle, it would then be associated with victory.

Although it would operate outside the player's conscious awareness, the pendant would be a symbol that takes on more and more emotional associations, thereby adding emotional depth to the story. However, because the pendant also plays a role in gameplay, it's doing double duty as a usable symbol.

Game example: Visual and verbal. In MAX PAYNE, above the rough-and-tumble squalor of the city float billboards for the mysterious Aesir Corporation, with its logo (the *r* in Aesir has a little wing on it) and its slogan, "A bit closer to heaven."

At first, the billboards have the emotional quality of taunting the residents of the city by reminding them of class distinctions. After Max (the main character) discovers that the Aesir Corporation is responsible both for the city's decrepit condition and the murder of his wife and child, the logo and slogan become associated with the enemy. And when Max triumphs in the end and finally attains some inner peace, he adopts the slogan "A bit

closer to heaven" as his own. The phrase is now associated with transcendence.

If this symbol only made MAX PAYNE players think about these different associations, then despite the fact that it was a wonderfully bold and inventive attempt, it was, to a great degree, unsuccessful. But if it evoked in players a variety of emotions that accompanied these different associations, then it was successful.

Going Deep

his article has covered four distinct techniques for evoking emotional depth with symbols. Each use is quite different from the other, and they can be used in combination. If no one notices your work after it's done, that's just fine — in general, they're not supposed to notice.

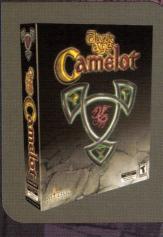
When using symbols, you're not creating intellectual exercises for your audience, forcing players to try to figure out what a symbol means. Using a symbol for that kind of mind game would detract from any emotional impact. Instead, when you use one or more of the techniques presented here, you deepen the player's emotional experience in the game by letting the symbol evoke the player's emotions.

While many of the examples of these techniques come from film, their use in games presents a unique tool to designers in the form of usable symbols functioning in gameplay. Games with stories have come far, but still have a distance to go. When game designers and writers master techniques to create complex characters and artfully evoke emotions during cinematics and gameplay, this new entertaining art form will truly have come into its own.

ACKNOWLEDGEMENTS

Thanks to Wagner James Au, David Perry, Chris Klug, Jason Bell, Henry Jenkins, Mike Morhaime, and David Taylor for their very helpful feedback in preparing this article.

Mythic Entertainment's DARK AGE OF CAMELOT



GAME DATA

PUBLISHER: Mythic Entertainment/Abandon Entertainment/Vivendi Universal Interactive Publishing

NUMBER OF FULL-TIME DEVELOPERS: 25

NUMBER OF CONTRACTORS: 5

ESTIMATED BUDGET: \$2.5 million

LENGTH OF DEVELOPMENT: 18 months

RELEASE DATE: October 9, 2001

PLATFORMS: Windows 98/ME/2000/XP

DEVELOPMENT HARDWARE (AVERAGE): 900MHz

Pentium IIIs

DEVELOPMENT SOFTWARE: 3DS Max, Photoshop, Visual C++, Linux GNU C++, various proprietary in-house tools

NOTABLE TECHNOLOGIES: NetImmerse, Linux open-source server and database products

ARK AGE OF CAMELOT was the best-selling computer game in the United States for the week of October 7, 2001, and is still comfortably in the top five as I write. This Postmortem is an overview of how this successful title was conceived and developed. My role on the project was as the game's producer.

Mythic Entertainment has been developing online games as a company since 1995 — forever in this field — but the company's founders had made online games even before then. In fact, as a company, we probably have more experience than any other company in developing online games of all types — over the years we have developed role-playing games, first-person shooters, top-down spaceship shooters, and strategy games. When I last wrote a Postmortem here in the pages of *Game Developer*, it was back in May 1998 for ALIENS ONLINE, our online first-person shooter based on the well-known *Alien* movies.

After ALIENS ONLINE, a nonaccelerated game, we created our first 3D-accelerated game, SPELLBINDER: THE NEXUS CONFLICT. During that project, we developed a relationship with NDL, makers of the NetImmerse 3D engine API toolkit. We learned a lot about 3D engine development over the course of that project and became very comfortable with software and art development in this environment. We finished SPELLBINDER, which went on to be a mildly successful Internet shooter, and it still has a small but loyal following.

After completing the SPELLBINDER project, we decided to create a graphical online role-playing game to compete with the thennew wave of online RPGs such as ULTIMA ONLINE and EVERQUEST, which were tak-

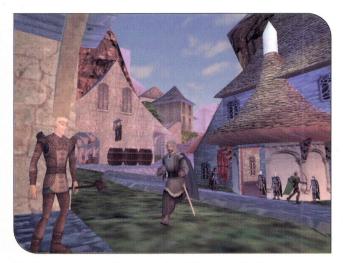
ing traditional text-based games and adding a graphical front end, with very successful results. Over the years, we had developed several nongraphical online role-playing games, including DRAGON'S GATE and DARKNESS FALLS: THE CRUSADE. Because of our experience developing RPGs, we knew that we had to have a slightly different slant on our new title in order to distinguish it from the RPGs that were already on the market. DARKNESS FALLS: THE CRUSADE (DFC) featured a built-in player-versusplayer (PvP) conflict in which three different teams, called Realms, fought each other for control of magical artifacts, known as Idols. We really liked this concept, which served to keep DFC players hooked on the game — especially because no other online game featured such team-based conflict as a core part of the game design. So, in late 1999, we decided to make a graphical version of DFC. The project was dubbed "Darkness Falls 3D," and we began preliminary work researching client engine and serv-



er technology.

MATT FIROR | Matt has been producing online games since the infancy of the industry. He has produced more than a dozen online games, including SILENT DEATH ONLINE, ALIENS ONLINE, ROLEMASTER: MAGESTORM, and of course, DARK AGE OF CAMELOT. He splits his time between a horse farm in Hunt Valley, Md., and a tiny apartment in Arlington, Va. Matt can be reached at mattf@mythicentertainment.com.

www.gdmag.com



The NetImmerse graphics engine from NDL proved flexible, stable, and packed with features players expect. Cities, creatures, world objects, and spell effects were created entirely in 3DS Max and exported using NetImmerse's Max plug-in, MaxImmerse.



CAMELOT's engine and client/server technology proved remarkably stable, displaying spell effects and combat animations, parsing system messages and chat, dispensing quests and tasks, and sending countless client/server messages with minimal effect on gameplay.

Right off the bat it was obvious that we had two major factors going in our favor. First, we determined we could use a much-enhanced version of the SPELL-BINDER graphics engine as DFC3D's client, just as we were able to use DFC's server code as a platform for the new game's back end. Having such a solid client and server right at the start — with associated client/server messaging alone saved us at least a year of development. Second, and even more advantageous, DFC's server came with that game's database of objects, monsters, and weapons. Indeed, we went into the CAMELOT project with a huge head start.

We were proceeding along under the DFC3D concept until our president, Mark Jacobs, came up with the idea of basing the game, at least partially, on the Arthurian legends. It was a great idea, since the stories of King Arthur are in the public domain, which meant we could use them with no fear of licensing issues. Of course, because the game was based on the idea that three Realms were in conflict, we quickly came up with the idea of basing the other two Realms on Norse Viking myths and Celtic Irish legends, respectively. Having the myths and legends of three cultures gives CAMELOT the feel of being three games in one, since each Realm has different races, classes, guilds, terrain, and monsters.

Because everyone knows what happened in Arthurian England, we based the game after Arthur's death and developed a back story of conflict among the three Realms. The game was rechristened DARK AGE OF CAMELOT, and around January 2000 we began the project in earnest. A year and a half and untold numbers of Monty Python jokes later, we finished the game.

The initial versions of DARK AGE OF CAMELOT used the rights for a tabletop role-playing game called Rolemaster as a basis for the class and spell systems. Not long into the project, the company that created Rolemaster, Iron Crown Enterprises, filed for bankruptcy, and we lost the rights. This turned out to be good for us, however, because we were no longer required to adhere to a set of rules based on the license — although we did have to scramble for about a week to rename and retune spells and classes and otherwise clear Rolemaster content out of the game.

As a company, Mythic had never before been able to devote all of its resources to any one game — we'd never had a project big enough to pay for it. Because of the sheer size and scope of CAMELOT, we

wanted to ensure that everyone at Mythic devoted themselves fully to the project. Doing so required an influx of money, and that's where New York's Abandon Entertainment stepped in. Abandon owns a couple of small companies, each of which specializes in different types of entertainment: a film studio, a web company, and a couple of game content development companies. Abandon wanted to become more involved in game development, so it purchased a minority stake in Mythic. This money allowed us to devote everyone on staff to the CAMELOT project, while also expanding and hiring much-needed programmers and artists. Our spreadsheets showed that we had enough money to support exactly 18 months of development starting from January 2000, giving the project a hard end date of September

By the summer of 2000, we had nearly our entire team in place. We had about 25 developers working full-time on the project — quite a small number compared to other online RPGs, but our existing technology allowed us to reduce substantially the amount of technical programming staff required. We had five programmers, ten world developers, seven artists, and several other people working on the game.



It all starts with a concept. The troll, a playable race, changed the most over the course of development from a hulking, human-like creature the more mythologically inspired version seen here.

Rob Denton, Mythic's vice president and chief technical brain, was responsible for all client and server programming, as well as the client/server messaging that tied the two together. His input was critical during design discussions, as he could tell us whether an idea would work or not. He immediately categorized features into "doable," "not doable," and the dreaded "on the list," which meant that it could be done, but he wouldn't commit to it. Brian Axelson was in charge of server programming as well as design of the game's combat system — a critical component in a PvP-centric game. Jim Montgomery provided CAMELOT's client interface coding and also designed and coded the game's magical spell system.

CJ Grebb and Lance Robertson led the art team. CJ was responsible for the game's look and feel, while Lance handled figure modeling and animations and managed the team's deadlines. Their team used 3DS Max and Character Studio to create CAMELOT's character and monster models and animations. The character models were technically advanced, as each in-game character has several different parts buried in it that can be turned off and on by the game. So, each model

can have a helmet head and a regular head (with hair) without having to load in a new model. Mike Crossmire created the game's spells in 3D Studio, tweaking the NetImmerse system to display animated spells with spectacular results.

The other major group in CAMELOT's development was the world team, led by Colin Hicks. This group was responsible for quests, monster placement, object placement, and just about everything else having to do with creating the world of DARK AGE OF CAMELOT. CAMELOT'S economy was designed by Dave Rickey. This economic system ensures that players must continue to spend money as they rise in level, which limits the amount of money that stays in the game. Dave and Mark Jacobs designed CAMELOT's trade skill system, which enables players to make armor, weapons, and other objects in the game - all tied to the economic system.

Among the myriad tasks that I did as a producer (writing, designing, persuading, arguing, and such), my job was to make sure all the teams worked together. I hosted an almost-daily morning meeting (at the wretched hour of 8:30 A.M.) where Colin, Rob, CJ, Lance, and I got together to make sure that we were all on the same page. I was also responsible for maintaining the master game client — all files added to the game had to be given to me, so I could verify they worked and then integrate them with the rest of the game.

For the game's sound and music, we contracted with Womb Music, based in Los Angeles, which had provided music for some of our previous titles. Rik Schaffer, the main guy at Womb, composed a wonderful soundtrack that consisted of several long main scores, as well as many shorter pieces in the style of Celtic, Norse, and old English folk songs, adding a sense of depth and quality to the world.

What Went Right

Community management/
beta program. From the
beginning of the project, we knew we

had precious few dollars available for marketing, and that our best chance to capture public attention would be to have a big presence on the various roleplaying fan sites around the Internet. One, the Vault Network, provided us with some message board space, a news page, and a couple of moderators, and we were off and running.

We devoted a lot of time over the year and a half that DARK AGE OF CAMELOT was in development to interacting with the future fans of the game. We hired a community relations manager whose sole job was to read different message boards and report back to us what was happening in the community. From the beginning, we took our fans seriously and made many tweaks and additions to the game based on their commentary and ideas.

No bureaucracy. Since the founding of Mythic, we have striven to have little bureaucracy. We have no levels, no directors, and few managers. We have a president, a vice president, and a producer. That's it for management, although for CAMELOT we did have to assign a lead world developer and art co-leads, just to streamline the day-to-day processes of the project.

Because of this simple command chain, we experienced no power struggles. We



POSTMORTEM

feel this is the best way to make a solid, cohesive game — a small group controls what the game is and how it is presented to the user. Because of this approach, decisions are made quickly, and features can be implemented without an endless line of approvals and politics.

Smart business decisions. Our close relationship with Abandon Entertainment was a critical factor in the success of the game. Abandon's purchase of a minority interest in Mythic ensured that we had enough money to fund the game from start to completion. Abandon's management was smart enough to realize that we knew more about game development than they, so they largely left us to make game-related decisions ourselves. They were involved in the project, of course some Abandon employees even became avid beta players of the game, even though most had never played an RPG before. Abandon's investment meant that we did not have to rely on any outside influence in designing or creating the game, which means that CAMELOT is wholly ours.

With Abandon teaming with us, Mark Jacobs, our president, decided to take a big chance and wait until the game was

ns.
h
cal

almost complete before looking for a distributor. In most cases, game companies seek out publishers, which typically have a hand in the design and production of the game and then distribute the game to the retail chain. With Mark's gamble, we produced the game ourselves (with critical financial help from Abandon and business advice from our business development person, Eugene Evans) and then looked only for a retail distributor. This gamble could have placed us at the end of the project with a great game but no way to get it into the hands of our customers. It all worked out in the end, of course, with Vivendi Universal stepping in and distributing — but on our terms.

Sweet serendipity. The CAMELOT project was helped immensely by factors completely out of our control — in other words, blind luck. Several high-profile online RPGs that were slated to launch at about the same time as CAMELOT were either pushed off (SHADOWBANE) or canceled outright (DARK ZION, FALLEN AGE). Also, the week we launched was originally scheduled to be the same week as the launch of WAR-CRAFT III, which will almost certainly be a huge seller. That project was also delayed, which ensured that CAMELOT launched as the only large-scale game, and the only online RPG, when it debuted on October 9, 2001. This little bit of good fortune gave the game a big initial boost, as there was little direct competition from other new products.

The joys of open source software and stability. Long ago, during the development of our early titles, we decided to use Linux wherever possible as our server back-end OS, and we kept to this same practice when creating Dark Age of Camelot. We have extensive Linux experience in-house, and it made sense for us to stay with a platform that we knew could handle the task and also was, well, free.

Because running CAMELOT would require a considerable amount of data



Balancing the races and classes for effective and challenging playerversus-player combat became one of Mythic's greatest challenges.



It was essential to provide players with plenty of player-versusenvironment conflict, such as with the forest giant seen here.



A look at the final version of the Troll. Every race within each Realm was designed to wear the same clothing, so the chain mail seen on this troll had to work equally well on the more diminutive Dwarven race.

management, we initially planned on using Oracle to store account and character information. However, Oracle's quoted license fee of more than \$900,000 quickly removed them from contention. Once we got over our shock and amusement at Oracle's pricing, we turned to a Linux-based freeware solution, MySQL, to manage CAMELOT's data storage, which so far has worked admirably.

Everyone developing games should at least investigate open source solutions for their servers. It's saved us a pile of money and has been stable and reliable. In fact, prior to CAMELOT's launch, it was axiomatic that MMORPGs were unstable and prone to crashing during their first month or so. From the outset, we were determined to buck this trend. We co-located our servers directly at UUNET, on the network backbone, which ensured a wide network pipe to the Internet. With this Internet connection, we can increase our bandwidth with just a few hours' notice to UUNET.



In addition to designing CAMELOT's many outdoor areas, Mythic's world development team had to populate those areas with interesting encounters and dynamic quests — no small task, considering they had not one but three distinct Realms to accommodate, as well as a finite amount of creatures available to them. Work on this content is ongoing, with new updates added to the game on a regular basis.

With the combination of reliable server code and a stable Internet connection—all running on open source software — CAMELOT went live on October 9, 2001, with virtually no problems. That first night, the game went down for about an hour and a half due to a database configuration problem, but since then, the game has been remarkably solid and stable. As of this writing, it hasn't been down due to server error for more than a few minutes ever since the first night.

What Went Wrong

Development of customer service tools. We really tried to avoid the customer service problems that are characteristic of some recently launched online games. One of the most important factors in keeping customer service reasonably effective was a smooth launch. Obviously, giving players fewer problems results in fewer calls to customer support. We did an excellent job with the launch — it went very smoothly.

However, we could have better foreseen other parts of our customer service plans.

First, we had a lot more players in the first week after CAMELOT went live than we ever could have forecast - 51,000 boxes were sold in the first four days alone. Our forecast numbers called for a much smaller number, and we hired our customer service staff based on this smaller number. Also, we put off creating customer service tools until much too late in the development cycle — some had yet to be developed when the game went live. These missing tools really hurt the customer service staff and added to the time it took to help each player with in-game problems. Eventually, wait times became much too long, and customer support as a whole suffered because of it. As I write, we still are trying to work ourselves out of this hole.

Lack of a cohesive marketing plan. We went into the CAMELOT project with a lot of experience in developing software, but no real experience in creating a marketing plan. We got

49

a lot of help with advertising from Abandon Entertainment, but there was no overall project plan. Basically, we took out ads in magazines that we thought were important and tried to keep on top of the Internet community. We didn't regularly issue press releases nor attempt to do a press tour or invite reporters to the Mythic offices to show off the game.

It's difficult to gauge just how much this hurt us. Our focus on Internet marketing gave us strong support among fans of the genre, but our lack of commercial marketing kept our company profile low, and we never received much mainstream media coverage because of it. Fortunately, we made up for our slow start, and then some, by our successful presence at E3. Abandon funded, designed, and staffed a large booth for us at the show, complete with medieval motif and lots of giveaways.

O Dungeons and Cities, where art thou? The first major update we made to CAMELOT's graphics engine to differentiate it from SPELLBINDER was to put in the rolling terrain system that makes the world so lifelike. We spent a long time making the outdoor areas of the game beautiful and well stocked with monster encounters. The ease with which we did this gave us a false sense of security when it came to developing our dungeon/city technology.





Creatures were modeled and mapped using 3DS Max and animated with Character Studio. Rumors that this zombie is a portrait of the producer after too many meetings are totally unfounded.

These areas in the game required a large number of models and characters in a much smaller space than the outdoor terrain, so creating dungeons and cities proved to be a much more difficult job than we thought. Because we put off doing the technical designs for the interior spaces for so long, in the end we simply didn't get enough of them done. The game launched with only three capital cities (one per Realm) and about 15 dungeons.

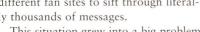
We have a great game but no servers! In a

great "Why didn't they tell us about this in college?" situation, we went into the final months of the project with no credit rating. Mythic Entertainment has been around for a long time, but we simply hadn't ever borrowed any money, and so we didn't have a credit history. This turned out to be a problem when we went out to lease our servers from Dell and were flatly denied. We pointed out that we had plenty of money in the bank, but to no avail. Dell simply wouldn't lease us the computers until we had a credit history. In the end, we were forced to purchase the servers outright from Dell, which obviously had a much greater impact on our bottom line.

Postrelease fan communica-• tion. As good as our communication with CAMELOT's fan base was during the game's design and beta periods, it began to suffer soon after the game's release. The community simply grew too large to communicate with in the manner we had during beta, when we simply went out to Internet message boards and posted our thoughts and plans. With the game live, it was obvious we needed a much more coherent way to communicate with our fans, one that would not send them to numerous

different fan sites to sift through literally thousands of messages.

This situation grew into a big problem when players became extremely frustrated by what they perceived as a lack of communication from us. About six weeks after release, we realized that we needed to create our own web site to publish information about the game: release notes, plan files, server status, Realm War status, and many other little things that we knew but our players didn't. This web site, dubbed "Camelot Herald," launched the following week and so far has been a great success. Fans of the game can now go to one web site to get all the information about the game in one place and with no interference.



For the Ages

t was a great pleasure to create DARK AGE OF CAMELOT, as it is the first big title that Mythic Entertainment has ever worked on. It was a wonderful thrill to see our names on top of the best-seller lists for those couple of weeks in October 2001, and we hope to be working on the game for a long time to come. As long as players are interested in playing the game, we'll be there adding content and updating it.

CREATIVECAREERS



Remember when games were fun? They still are at Blue Shift!

If great games are in your future you belong at Blue Shift.

We have immediate openings for programmers, artists & animators.

Blue Shift is located in the heart of Silicon Valley.

Close to Stanford

University. Steps from CalTrain, fabulous restaurants, cafes, and more...



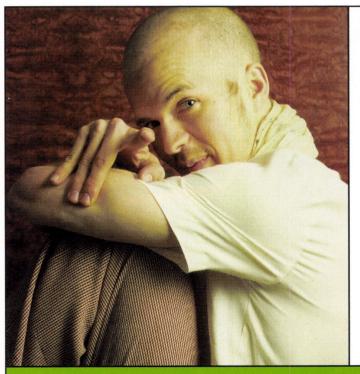
Come join us. And have some fun!

Siff Email your resume to:

Jobs@blueshiftinc.com

Or FAX it to us at: 650.614.4856

200 S. California Avenue, Suite 201 Palo Alto, CA 94306 650.614.4855 www.blueshiftgames.com



Gamasutra.com

It's simple:

Search thousands of resumes by profession, game credits, technical skills, or preferred employment status.

Find experienced, professional game developers with an average of 5 years of industry experience and more than 4 titles developed.

Perform unlimited searches during your access period.

GamaNetwork

How to hire a game developer.

www.gdmag.com





SEE WE

LIKE YOU'VE NEVER SEEN IT BEFORE

Introducing an amazing feat of compatibility...what you've all been waiting for Maya on the Mac OS X!!
Welcome to the family.

CAN YOU IMAGINE"

With the full family of Maya products to its name, Snader and Associates provides unparalleled service and support for gamers. For over a decade, we have been the leading Alias|Wavefront solutions provider. Contact us today and let Snader and Associates simplify your life.

Call 415.257.8480 or email us at games@snader.com



Alias wavefront™

Starting at \$7500 USD. Available on Windows NT™, Windows 2000™, IRIX®, Linux®, and Mac®OSX™. To find out more about the most powerful 3D software money can buy, visit www.snader.com

TARGETPAVILION

GETEDUCATED

Your project is awesome.

Demand the best audio.



music • sound • voice



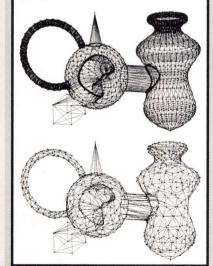
415.454.8988

www.thesounddepartment.com

POPLID 2.0 Polygon Optimization Library

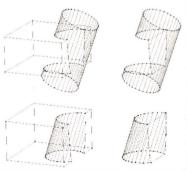
POPLib 2.0 is available as C++ Source Includes: Tessellation; Decimation; Booleans; Sectioning; Ray Firing; and other Polygon and Geometry Tools Great tools for Software Developers!

Decimation



+1.858.538.3800

Booleans



www.PolygonLibrary.com Click Here for Fewer Polygons! Click Here for Better Polygons! Click Here for Faster Polygons!

info@PolygonLibrary.com







It's Alivel, Alias/Wavefront Maya, dobe Photoshop
Deep Paint 3D and Adobe After Effects
Terry Ziegelman, 5T. LOUIS, MO., SECOND YEAR M.FA.
Paul George, VALENCIA, VENEZUELA, SECOND YEAR M.FA.
Stephen Johnson, CHARLESTON, S.C., B.F.A.
Jamie Kirschenbaum, RESTON, VA., B.F.A

BACHELOR OF FINE ARTS
MASTER OF ARCHITECTURE
MASTER OF ARTS
MASTER OF FINE ARTS

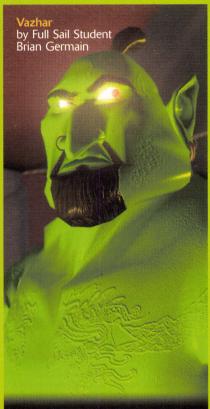
Architectural History Architecture Art History

Computer Art

Fashion
Fibers
Furniture Design
Graphic Design
Historic Preservation
Illustration
Interior Design
Media and Performing Arts
Metals and Jewelry
Painting
Photography
Product Design
Sequential Art
Video/Film

AN INTERNATIONAL UNIVERSITY FOR THE ARTS
Savannah, Georgia
800.869.7223 | www.scad.edu

GETEDUCATED





School of:

Game Design Computer Animation Digital Media Film Audio **Show Production**

800.226.7625 www.fullsail.com

3300 University Boulevard Winter Park, FL 32792

- · Financial aid available to those who qualify.
- Job placement assistance.
 Accredited by ACCSCT.

©2001 Full Sail, Inc. All rights reserved. The terms "Full Sail," "Full Sail Real World Education," and the Full Sail logo are either registered service marks or service marks of Full Sail, Inc.





continued from page 56

If we honor by title, then we risk award sweeps, as one fad or another dominates in any given year. If we honor by artist and engineer, we stand a chance of delving beneath the surface far enough to acknowledge important, if less trendy, work. It doesn't necessarily follow, for example, that the best game of any year has the best sound, or the finest animation, or the most capable engine.

I cast another vote for simple integrity. Awards should be as free from politics as possible — otherwise they don't mean much. Publishers, manufacturers, and allied companies have their PR machines, their marketing agendas, their bottom lines. Self-interest is built into their charters of incorporation. Awards should not contribute to their further aggrandizement, whether by naming them as recipients or by allowing them to sponsor our ceremonies.

To further ensure community confidence in our awards, we also need to improve our procedures. There's a natural tension between art as craft and art as experience. Here, Hollywood seems to have hit upon a satisfying compromise: Oscar nominations are made by peers, and the final awards are voted upon by all. We should follow this practice. Level designers are the only developers qualified to identify the best levels, for example, yet the rest of us can readily judge the fun factor among selected nominees.

If we work at it, the result will not only be better games, but clout. Hollywood, home of the rudest pop entertainment, has become immune to unfair pressure simply by declaring, through its web of awards, that movies are an art form. Establishing our own well-conceived awards should help protect us against the slings and arrows of outraged congressmen and social busybodies who imagine

that good art is like good nutrition — the five food groups of character formation, as it were.

It took decades for the Oscars to become the show-biz phenomenon they now are. Yet Frank Capra's win for directing 1934's It Happened One Night is as well remembered as Steven Spielberg's award for 1998's Saving Private Ryan because the process was solid from the beginning. Similarly, it may take a while for game awards to acquire public cachet. But to developers, those honored and those voting, the benefits are immediate and lasting.

leader at LucasArts, where he is working on a new console title. He is also a member of the Academy of Motion Picture Arts and Sciences. He was given a Spotlight Award for Indiana Jones and the Fate of Atlantis at GDC many long years ago.

ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE	COMPANY NAME PAGE
2015	51	House of Moves	C2	Savannah College of Art 53
Analog Devices	15	Integrity Ware	53	Seneca College 54
Blue Shift	51	Metrowerks	5	Snader and Associates 52
Compaq Systems	21	Microsoft Developer	3	Softimage C3
Core Microsystems	17	Numerical Design	13	The Sound Department 53
Criterion Software	7	Numerical Design	13	The Sound Department 55
Full Sail Real World	54	RAD Game Tools	C4	Vancouver Film School 54

www.gdmag.com 55

Illustration by Deborah Stephens

The



et's imagine games are an art form. I know, I know
— for many of us in contact with the so-called real
arts, the notion sounds pretentious. It also makes
developers who are former computer science
majors edgy, because it challenges assumptions
that games are founded upon technology. Still, it's a useful con-

that games are founded upon technology. Still, it's a useful concept. It's especially useful when we start to think about the mediocre state of our profession, and about ways to elevate our aims, aspirations, and attitudes.

Art is what people accomplish when they don't quite know what to do, when the lines on the road map are faint, when the formula is vague, when the product of their labors is new and unique. Sound familiar? This is the everyday challenge facing game developers: create something new and unique. Incredibly, we often succeed. The real problem is, new and unique isn't enough — most of us also want to build games that are actually good. Good by any standard. Good today, better than yesterday, and worthy enough for tomorrow. Good even when we can't exactly define what "good" means.

How can we focus our energies on such a lofty and elusive goal? It's tough enough to focus on shipping our next title. The

Envelope, Please

best method I know comes directly from some of those "real" arts: the annual round of awards when movie, television, and music academies honor their members' achievements of the previous year.

I believe that awards are an inspiration to all of us — whether we're ever nominated or not. Seeing our colleagues honored raises our sights and ambition above the petty requirements of the marketplace, and also above the dismal recognition that comes from what passes for a trade press in our business. Awards mark the framework in which a consequential meta-discussion about excellence takes place among game developers — buzz translated into votes.

The process is already in place, and while some newer awards programs try to improve on the problems of the older ones, they nevertheless generate struggle and controversy. Should we honor titles or people? Developers or publishers? Should we accept sponsorship? Should we control the nominating process? Should we aim for a marketable entertainment package with our award ceremonies? These issues are important. Establishing a firm basis for our awards will contribute to industry growth and maturity.

I hereby cast a vote in favor of maximum exposure. Awards should discover and celebrate as many of the arts and crafts of game development as possible. We need to spread far and wide the idea that individual human beings are responsible for the games we play. We may spin idle dreams about theoretical possibilities, but what spurs us to action are real achievements, against all odds, by real people toiling in the real world.

continued on page 55

"NOW, WHERE'S THAT FINAL FRONTIER?"



SOFTIMAGE® XSI® v.2.0 GETS YOU THERE.

XSI v.2.0 takes you beyond your boundaries. Imaginary. Or real. The newest version of XSI allows you to explore advanced dynamics with the seamlessly **integrated photo realistic hair/fur** simulation system. Discover the industry's only fully-interactive renderer, powered with **mental ray**® v.3.0. And broadcast artists will find the super sophisticated 3-D text and logo creation tools a dream come true.

Want to fly further still?

How about an awesome new integrated **compositor** to unite your 2-D and 3-D worlds with unprecedented speed and elegance. And our revolutionary **interactive real-time shaders** give artists the power to control game effects. All with mind-numbing performance and stability. Not to mention accelerated ROI.

Want to **impress aliens** – or even your boss? Visit **www.softimage.com/v2** where you can order the very hot **XSI demo CD**, sign up for the XSI newsletter – even arrange for a hands-on demo. Suddenly, that **final frontier** seems a whole lot closer.



Innovate. Create. Collaborate.





AVIC.

Make, Manage and Move Media*

The Best in Game Development Technology



GRANNY 3D ANIMATION

The most sophisticated dynamic 3D animation system. Smoothly animates your characters - blends from a run to a walk to a sliding stop, all while using real-time inverse kinematics to point or look at other objects. Supports vertex buffers and display lists for blazing speed! Deforms clothing to fit different characters. Includes built-in Bink and S3TC texture compression. Exports perfectly from 3D Studio Max and Alias|Wavefront's Maya. Get dynamic models into your game today - use Granny.



BINK VIDEO TECHNOLOGY



The best video codec - make your videos shine! Unrivaled quality (better than DVD and MPEG II). Up to three times faster than other true-color codecs. Perceptibly lossless, 8 to 1 audio compression. Decompresses to DirectDraw, DIBSections, YUV overlays, or any other piece of memory. Available for Win32, MacOS, and now Xbox. Think Bink!



SMACKER VIDEO TECHNOLOGY

The best 256-color codec on the planet! Smacker is the codec of choice for **256-color games** (of course), games targeting Pentium 133s and below, games with video sprites or video transparency (**much faster in 256 colors**), or games with extremely high-resolution videos (800x600 and up). Smacker is available on Win32, Win16, DOS, and MacOS.



MILES SOUND SYSTEM



The ultimate sound system! Miles supports 2D audio, 3D audio, MIDI with DLS, streaming, cd audio (red book), **DSP filtering**, **MP3 playback (patent rights included)**, **Internet voice chat** (2900, 2400, and 1200 bits per second codecs), Creative **EAX 1 and 2**, Aureal **A3D 1 and 2**, RSX 3D, DirectX 7, Dolby Surround, QSound, **software EAX emulation**, and more!



425 893 4300 www.radgametools.com

POWERFUL TECHNOLOGY. EASY TO USE. NO ROYALTIES.